

**Optimizing implicit plant interactions to conserve water on
residential landscapes**

by

Rhonda Hoenigman

B.S., Ohio University, 1994

M.S. San Diego State University, 2007

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science

2012

This thesis entitled:
Optimizing implicit plant interactions to conserve water on residential landscapes
written by Rhonda Hoenigman
has been approved for the Department of Computer Science

Elizabeth Bradley

Nichole Barger

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Hoenigman, Rhonda (Ph.D., Computer Science)

Optimizing implicit plant interactions to conserve water on residential landscapes

Thesis directed by Elizabeth Bradley and Nichole Barger

In drought-prone environments, interactions between plants can enable individuals to conserve water. The physical location of each individual not only influences its resource needs, but also affects the availability of resources on the landscape. This thesis seeks to model this observed behavior in natural systems and use that model to create a new optimization approach for water conservation in residential systems in drought-prone climates. In this research, the arrangement of plants on a landscape is treated as a location optimization problem where the objective is to find the best locations for a given set of plants on a landscape with a given set of resources.

The biological properties of the domain make this optimization problem novel. The search space is influenced by what I call *neighbor effects*, which include heterogeneity, locality, and feedback. Plants are heterogeneous, both in their requirements for growth and in their effects on their surroundings. Interactions with other plants are primarily local, and can either increase or decrease growth depending on the types of plants involved and the resources on the landscape. These interactions also mean that distance is best treated *implicitly* in modeling the effects of the interactions, rather than *explicitly* as it is in most location problems. Both heterogeneity and locality generate feedback conditions, whereby the positioning of a plant on a landscape changes the resources available at that location. The placement of a tree, for example, changes the light available nearby, which then affects the growth of any plants located in those modified conditions.

The model presented here captures the domain-specific features of this problem using an agent-based strategy. In this model, each plant is an agent that “lives” on a simulated landscape. Agents have light and water requirements for growth, selected to match the requirements of real plants, and a growth curve that determines agent fitness in simulated light and water conditions. A multiobjective fitness function captures the trade-off between maximizing plant agent growth on

the landscape and minimizing each agent's water use. The total water needed on the landscape is the emergent property of this individual behavior.

Two optimization strategies commonly used on location problems — simulated annealing and a genetic algorithm — are applied to find the locations on the landscape that maximize the fitness score. These strategies are compared to an agent-based optimization routine that is designed to capitalize on the novel features of this domain. In this approach, agents maximize their own individual fitness instead of the global fitness on the landscape. This distributed strategy produces solutions comparable to the genetic algorithm, but in a fraction of the time required for that algorithm.

This work also includes experiments with live plants to generate the data for the agent growth curves and to validate the performance of the optimized arrangements. The growth curves were generated using two species with different light and water requirements grown under a range of light and water conditions. A validation experiment, where live plants were arranged in a random and an optimized configuration, shows the value of this optimization approach for water conservation. Both arrangements had zero mortality and all plants looked healthy throughout the experiment. However, the optimized arrangement used significantly less water.

Dedication

Without a doubt, I would like to dedicate this dissertation to my family for their selfless acceptance of this pursuit. I would also like to dedicate this dissertation to the 1000's of *Tagetes patula* and *Impatiens wallerana* who gave their lives for this work.



Acknowledgements

I would like to acknowledge my friends and family for their assistance, friendship, and support over the last five years. Without the support of my family, in particular, this endeavor would not have been possible.

I would also like to thank my advisor Elizabeth Bradley for providing guidance and mentorship.

My co-advisor Nichole Barger deserves special mention here. Interdisciplinary research that has a computer scientist voluntarily spending countless hours working with plants in a greenhouse is bound to be fraught with difficulty. Nichole provided considerable help and support and was always willing to listen to me complain about plants refusing to grow. I hope we have a chance to work together again and have many more conversations about science, education, and parenting.

Contents

Chapter

1	Introduction	1
1.1	What makes this location problem special	2
1.2	An agent-based model for optimizing landscape performance	3
1.2.1	Optimization strategies	4
1.3	Experiments and results	5
1.4	Thesis and contributions	6
2	Modeling landscape resource properties and individual plant growth	8
2.1	Plant water use basics	9
2.2	Single-species models	13
2.3	Multi-species models	14
2.4	The need for multi-species optimization	15
3	Location optimization	17
3.1	Location problems	18
3.2	Multiobjective optimization	19
3.3	Other combinatorial optimization problems	21
3.4	A different kind of location problem	22
3.4.1	Neighbor effects	23
3.5	Optimization algorithms	24

3.5.1	Metaheuristics	25
3.5.2	Agent-based methods	33
4	Agent-based modeling	35
4.1	Determining properties and behaviors	37
4.2	Agent interactions	38
4.3	How models are used	39
4.3.1	Optimization	39
4.3.2	Decision support	40
4.4	Validation	41
5	Building the plant-growth model	43
5.1	Experimental setup	44
5.1.1	Determining how light levels affect plant growth	45
5.1.2	Modifying light on the landscape	47
5.1.3	Determining how watering frequency affects plant growth	48
5.1.4	Spacing and competition for water	49
5.2	Data analysis and filtering	50
5.3	Results	51
5.3.1	Germination	51
5.3.2	Growth response to water — marigolds	52
5.3.3	Growth reduction from water stress — marigolds	59
5.3.4	Growth response to water — impatiens	60
5.3.5	Growth response to light — marigolds	65
5.3.6	Growth response to light — impatiens	69
5.3.7	Spacing and competition for water	72
5.3.8	Water application efficiency	75

6	An agent-based model	82
6.1	Landscape	83
6.2	Plant agents	84
6.3	Plant growth model — water	86
6.4	Plant agents — competition	87
6.5	Plant growth model — shade	88
6.6	Water application efficiency	89
6.7	Calculating agent fitness	91
7	An agent-based search algorithm	96
8	Experimental methods	101
8.1	Optimization strategies	102
8.2	Landscape conditions and agents	105
8.3	Water application efficiency	108
8.4	Validating a simulation in the greenhouse	109
9	Results	113
9.1	Fitness scores	114
9.1.1	Evaluating individual plant-agent mortality	116
9.2	Fitness function α and β parameters and water application efficiency	120
9.3	How agents' watering frequency and location affected the emergent water applied .	126
9.4	Greenhouse validation experiment	131
9.5	The effects of optimization-strategy parameters on fitness scores and runtime . . .	135
9.5.1	Agent search	135
9.5.2	Simulated annealing	137
9.5.3	Genetic algorithm	137
9.6	Runtime comparison of the optimization strategies	141

9.7	Results from a commercial optimization package	145
9.7.1	Setting up the Matlab GA	145
9.7.2	Why using the Matlab GA is better than implementing your own GA . . .	146
9.7.3	Limitations of the Matlab GA	149
9.8	Future work	151
10	Conclusion	155
	Bibliography	157

Tables

Table

3.1	Plant placement matrix, where 1 indicates the placement of a plant and 0 is no plant.	29
3.2	4x4 landscape with six plants converted to a GA Chromosome. The GA Crossover and GA Mutation images here also show the process of crossover and mutation, which are how new solutions are created from the existing population.	30
5.1	Cell coverage in all marigold trays. These numbers reflect the percentage of cells where at least one plant germinated.	52
5.2	Cell coverage in all impatiens trays	52
5.3	Confidence intervals and p-values for mean growth in the three shade levels.	67
6.1	Fitness scores for marigold light, water, and spacing combinations. These scores are calculated using the model equations described in this chapter. The biomass values are from the greenhouse experiments described in Chapter 5. The Shade and Water columns are the light reduction and water inputs to the fitness equations, where $shade \in [1, 2, 3]$ is full sun, partial sun, and shade respectively, and $water \in [1200, 600, 400, 300]$ is high, medium, low, and very low respectively. The Space column describes whether plants were grown with all surrounding cells populated by other plants or by empty cells.	94

6.2	Fitness scores for the impatiens in the light and water combinations. Unlike the marigolds, additional spacing around each plant does not affect the fitness scores for this species.	94
-----	---	----

Figures

Figure

- 3.1 The position of one plant can affect other plants by modifying the light at each location. In this example, the tree is generating shade in surrounding cells, which changes the light available for the other plants. The shaded cells are shown in grey. 23

- 5.1 Tray with mesh grid used for planting. Each cell contained one plant, establishing evenly distributed, discrete locations for the plants in the experiment. 45
- 5.2 Experimental setup, including planting trays, mesh grid for controlling plant locations, and shade screen to generate variable light conditions. 46
- 5.3 Impatiens experiment showing the effects of shading on the landscape. Cells to the left of the shade screen were shaded and cells to the right of the shade screen received full sun all day. 47
- 5.4 Map of the light distribution on the landscape in the marigold experiments. Available light was determined by counting the shaded cells at multiple times throughout the day. Blue cells are those that received full shade, and red cells are those that received all day sun. Cells with colors between red and blue received a mix of sun and shade. 48
- 5.5 Apparatus used to weigh trays. Trays were weighed each day before and after watering to determine water loss over the previous 24 hours. 49

5.6	Marigolds planted in every other row and column in the competition experiment to determine how the additional space around each plant influenced its growth at different watering frequencies.	50
5.7	Setup of marigold competition experiment. Cells covered by the shade screen received full shade and the rest of the tray received full sun. Plants were spaced every other row and column.	51
5.8	Mean biomass and the 95% confidence interval for marigold trays as a function of the water applied to the tray. Growth increased with increased water application up to the every other day watering frequency. Beyond this, the increase in growth leveled off for the amount of water applied as plants could not use all the water that was applied.	53
5.9	Mean water loss (with 95% confidence interval) as a function of the water applied at all four watering frequencies. The very low, low, and medium watering frequencies (300 mL, 400 mL, and 600 mL respectively) trays lost all of the water applied. The high watering frequency trays, however, lost only 933 mL of the 1200 mL daily.	54
5.10	Tray weight over time for the high watering frequency marigolds. In these trays, the water applied was greater than the ET rate. As a result, the weight of the tray increased over time as the soil became saturated and water collected in the catch trays.	55
5.11	Total water loss each day in both high watering frequency trays. Each tray lost a mean of 932.7 mL daily: the mean for both trays was 1865.4 mL. There was no clear pattern in water loss over time to indicate that trays lost more water as plants increased in size.	57
5.12	The percentage of the maximum mean biomass produced in all marigold experimental conditions as a function of the minimum percentage of water consumed in a tray.	60

5.13	Mean biomass with the 95% confidence interval for the impatiens as a function of the amount of water applied. The x-axis in this graph is the mean daily water applied, where 1200 mL was the high watering frequency, 600 mL was the medium watering frequency, 400 mL was the low watering frequency, and 300 mL was the very low watering frequency. Unlike the marigolds, where growth increased with additional water applied, impatiens grew less in daily watering than when they were watered every other day.	62
5.14	Impatiens watered every day (high watering frequency) on the final day of the experiment. These plants did not perform as well as the impatiens watered every other day.	62
5.15	Impatiens watered every other day on final day of the experiment. This species performed better with watering every other day than with daily watering. This image also shows the effects of shading. Plants to the left of the shade screen that received shade or partial shade performed better than plants to the right of the screen, which received full sun throughout the day.	63
5.16	Total water lost each day for the impatiens in the high watering frequency tray. Just as with the marigolds, there was not a clear increase in water loss over time as plants increased in size.	63
5.17	Tray weight (grams) over time for low watering frequency tray. The weight of the tray decreased throughout the experiment, indicating that the soil was drying out. This decreasing weight also explains why ET was higher than the amount of water applied for the first 12 days of the experiment. There were existing water resources in the tray that were being depleted, and the weight of the tray decreased as the soil dried out.	66

- 5.18 Mean biomass with the 95% confidence interval for the marigolds in the three shade conditions (shade = 1,2,3 for full sun, partial sun, and shade, respectively) for each individual watering frequency. This graph shows that the reduction in growth due to light reduction depended upon how much water the plants received. When the plants were watered every day or every other day, they experienced more of a growth reduction in low light (shade = 3) than when they were watered less frequently. In the very water-limited situation (every four days), reducing light increased growth as plants in the shade benefitted from the reduced water requirements for survival in these conditions. 68
- 5.19 Mean growth vs. watering frequency in full sun, partial sun, and shade conditions. The x-axis here is the mean daily water that was applied, where 1200 mL was the high watering frequency, 600 mL was the medium watering frequency, 400 mL was the low watering frequency, and 300 mL was the very low watering frequency. In shade conditions, reduced water had less of an effect on growth than it did in full sun conditions. 70
- 5.20 Mean growth with 95% confidence interval vs. light for all four watering frequencies for the impatiens. The shade categories here are full sun, partial sun, and shade (shade = 1, 2, 3, respectively). Unlike the marigolds, these plants grew better in the shade than in the full sun. The effect of the light was the most dramatic in the medium watering frequency (every other day), which was the optimal frequency for this species. 71
- 5.21 Roots growing along the bottom of one of the trays for the high watering frequency marigolds. This image shows approximately half the tray. Clearly, these plants did not restrict their growth to the cells in which they were planted and surrounding cells 74

- 5.22 Mean growth in full sun when marigolds were planted in every cell in a tray and every other cell in a tray. In high watering frequency (watering frequency = 1), full sun conditions: increasing the space around each plant did not significantly increase growth. However, in full sun with medium, low, and very low watering frequencies (watering frequency = 2,3,4), the additional space did increase growth. This was particularly true in the very low watering frequency. 76
- 5.23 WAE in g/mL for the marigolds in all watering levels. The x-axis is the daily water applied to a tray for a watering frequency and the y-axis is the grams/mL. Although the marigolds had the highest biomass in the high watering frequency, WAE peaked at the medium watering frequency. This was the frequency that produced the most biomass for the water applied. 78
- 5.24 The WAE in g/mL for the marigolds varied for the three light categories for each watering frequency. In the high, medium and low watering frequencies, WAE was lower in the full shade (shade = 3) than in the full sun (light = 1). For the very low watering frequency, WAE increased in the full shade (shade = 3), possibly due to a 67% mortality rate in full sun as compared to a 4% mortality rate in full shade. . . 80
- 5.25 WAE in g/mL for impatiens for all watering frequencies. The highest observed WAE was in the low watering frequency for this species. However, the WAE for the medium, low, and very low watering frequencies were all similar. 81
- 6.1 Plant-agent and landscape properties. Plant-agent properties determine how the agent responds to, and influences, its surrounding. The landscape properties captures the effects of plant agents on the available light and water resources. 85

6.2	Parameters in the growth and WAE components of the fitness function. The growth score captures how well the light and water properties on the landscape match the agent's requirements for light and water. WAE uses the growth score and the properties on the landscape to determine how efficiently the agent used the water provided, i.e. how much did it grow for the water applied.	90
6.3	Correlation between the fitness scores in the model and biomass for marigolds and impatiens in the greenhouse experiments. The linear fits shown here have $R^2 = 0.76$ for the marigolds and $R^2 = 0.84$ for the impatiens. There is one outlier for the marigolds, which is the fitness score assigned to the plants grown in full-sun, very low watering conditions with surrounding empty cells. The model uses a conservative estimate for these conditions when plants have reduced competition from nearby plants. This conservative estimate means that the model underestimates fitness in these conditions.	95
8.1	Light distribution of shade, partial sun, and full sun conditions for simulations and live validation experiment. Blue areas are shade, green is partial sun, and red is full sun. The brown ovals show the positions of the simulated shade screens.	106
8.2	Watering frequencies on the landscape used for all numerical simulations as well as the live-plant validation experiment. The landscape includes all watering frequencies used in the greenhouse experiments; 1200 mL of water was applied at each watering.	108
8.3	Random arrangement used for greenhouse validation experiment. Plants were evenly distributed over the landscape, and each watering frequency contained a mix of each species. This arrangement was used for comparison to an arrangement of the same collection of plants that was optimized by agent-search for water application efficiency.	111

- 8.4 Arrangement optimized for water application efficiency used in the greenhouse validation experiment. This arrangement required 37% less water than the random arrangement of the same plants that is shown in Figure 8.3. In this arrangement, there were very few plants in the high watering frequency region: most plants for both species were in either medium or low watering frequency. This placement reflects the emphasis on water efficiency over growth in the fitness function. There were also only a few plants in the very low watering frequency region, which reflects the other trade-off in this problem — plants need water to grow. 112
- 9.1 Mean fitness scores for each optimization strategy for 20 to 100 plant agents (half from each species) on the simulated landscape with the light and water properties shown in Figures 8.1 and 8.2. SA consistently produced the highest fitness of the three strategies. However, the difference in the fitness scores for these strategies was less than 10% on all plant collections. 117
- 9.2 Percentage of the plant agents of the experiment in Figure 9.1 that experienced a reduction in growth due to competition for water with nearby agents. As more plants were added to the landscape, competition increased, lowering the average fitness of the plants. However, the increase was not linear. There was limited competition with fewer than 60 plants on the landscape, and the competition increased sharply between 60 and 80 plants. With 80 or more plants, all plants experienced a fitness reduction due to competition. 118

- 9.3 Mean fitness scores (with 95% confidence interval) for plant agents in the random arrangements on the landscape shown in Figures 8.1 and 8.2. In these simulations, a random location was selected for each plant agent and the fitness score was calculated at that location. Just as in the optimized arrangements, fitness dropped as more plants were added to the landscape. Scores were lower overall than they were for the optimized arrangements. “Mortality” (scores below the 0.45 survival threshold) on the landscape was also higher with random placements than for the optimized arrangements. 119
- 9.4 Mean watering frequency at the locations of all plant agents in the 20-plant SA solutions. This graph shows that increasing β decreased the required watering frequency for individual plant agents. There was a sharp drop in the watering frequency when $\beta \geq 0.5$, which places equal or greater priority on *WAE* over *growth* in the fitness function: $fitness = \alpha \cdot growth + \beta \cdot WAE$ 121
- 9.5 Mean watering frequency at the locations of all plants in the 30-, 60-, and 80-plant solutions for SA. This graph shows that increasing β decreased the required watering frequency for individual plants, and that the watering frequency and the effect of β also depended upon the number of plants on the landscape. With fewer plants, increasing β caused a sharp drop in the watering frequency. With 80 plants, watering frequency was stable until $\beta \geq 0.8$ 123
- 9.6 The mean number of neighbor competitors when 30-, 60- and 80-plant agent SA solutions. With 30 plant agents, the number of neighbors decreased as the value of β increased: there was room on the landscape for all plants in the lower watering frequency regions. However, as more plant agents were added to the landscape, competition for water increased due to the higher number of neighbor competitors in the 60- and 80- plant agent solutions. A high β value was needed for WAE to offset the growth reduction from competition. 125

- 9.7 Mean water applied (with 95% confidence interval) to the landscape for all β values for the 20-plant collection arrangements produced using agent search. The water applied to the entire landscape follows the same pattern as the watering frequency of individual agents: increasing β reduces water use for both the individual plants and the entire landscape. 127
- 9.8 Water applied vs. fitness for the 20-plant solutions for SA, GA, and agent search for representative β values. In the SA solutions, there was a clear delineation in the water applied for different values of β . For the GA and agent solutions, the pattern was more variable, but still showed a trend of reduced water and fitness with increasing β . The variability in the water applied for the same fitness score in solutions for all three optimization strategies was due to the effects of spacing on the number of cells to water and competition for that water. 128
- 9.9 Mean water applied (with 95% confidence interval) for all three optimization strategies. These plots show the difference in the water needed by the different solutions when the priority is on *growth* ($\beta = 0.0$) or *WAE* ($\beta = 1.0$). In the SA solutions, the curves for the two values of β diverge around 50 to 70 plant agents, where the optimization was most effective at reducing water use on the landscape. For the GA and agent solutions, prioritizing *WAE* also reduced water applied, but not as significantly as in the SA solutions. 130
- 9.10 Random and optimized arrangements for the validation experiment with live plants. The optimized arrangement was selected to conserve water, but to still achieve zero mortality throughout the experiment. 132
- 9.11 Random and optimized arrangements on the final day of the validation experiment with live plants. The optimized arrangement used 35% less water than the random arrangement but also produced equivalently less biomass. 134

- 9.12 Mean fitness (with 95% confidence interval) vs. moves allowed in the 20-, 50-, and 80-plant agent solutions produced by the agent search algorithm. In the solutions shown here, allowing agents to make even one move significantly improved fitness over the random placements. However, additional moves allowed did not produce additional fitness improvements. As the number of plant agents on the landscape increased, so did the number of moves allowed that were needed for the algorithm to converge to a solution. 136
- 9.13 Mean fitness scores (with 95% confidence interval) vs. cooling value for 20-, 50-, and 80-plant solutions for SA. Allowing the search algorithm to run longer — by using a higher value for the cooling parameter — produced higher mean fitness scores for each of the plant collections. 138
- 9.14 Mean fitness scores (with 95% confidence interval) for the 20-, 50-, and 80-plant collections for 300 to 1800 iterations optimized by the GA. The number of iterations determined how long the algorithm searched for a solution. Allowing the algorithm to run longer improved solution fitness, but this effect slowed at around 1300 iterations. 140
- 9.15 Mean runtime in seconds (with 95% confidence interval) for agent search, SA, and the GA for 20-plant solutions. Runtime was defined as the user CPU time to execute the algorithm on the Janus supercomputer. The agent search ran significantly faster than the other two algorithms while producing solutions with similar fitness scores. 143
- 9.16 Mean runtime in seconds (with 95% confidence interval) for all three optimization strategies for 50 plants. Again, agent search ran faster than the other two algorithms in these tests. 144

- 9.17 Mean fitness scores (with 95% confidence interval) for all plant-agent collections optimized using the Matlab GA with the default parameter settings. This results from this GA followed the same general pattern as the results from the author's GA implementation — fitness deteriorated as more plants were added to the landscape. 148
- 9.18 Fitness scores vs. number of plants using a GA encoding where the number of plants and plant types were not fixed. For most solutions, the optimized landscape included 30 and 60 plants. Fitness declined as more plants were added to the landscape, which was also observed in simulations where the number and type of plants were fixed. 150

Chapter 1

Introduction

Residential irrigation accounts for up to 70% of household water use in semi-arid and arid regions in the U.S. [41], where water is also a scarce resource. However, landscapes in these regions provide benefits—trees can reduce the heat index around the home, which decreases air conditioning use and saves energy [13, 96]. As a result of these benefits and the additional comforts that landscaping provides, there are efforts to reduce water use on residential landscapes. These efforts include xeriscaping, which encourages the use of drought-tolerant native plants [59], and irrigation systems that measure water loss or set application rates based on a microclimate index [23], i.e. hot, dry areas receive more water than shaded areas.

On wild landscapes in drought-prone environments, interactions have been observed that enable plants to conserve water. This natural phenomenon, known as *facilitation* [17], occurs when larger shrubs serve as benefactors to smaller annuals and perennials by generating conditions that protect them from harsh afternoon sun, enabling the annuals to survive on the scarce water available [47, 111].

These examples of facilitation illustrate how the physical location of individual plants on a landscape can affect the fitness of other plants in proximity. This thesis models this observed behavior in natural systems and uses that model to create a new optimization approach for water conservation in residential systems in drought-prone environments. The objective is to find the best locations on a landscape with a given set of resources for a given set of plants. The physical location of each plant affects the fitness of the individual as well as the fitness of surrounding individuals

through both positive and negative effects on available resources. With any arrangement, plants modify their surroundings by generating shade and consuming water. These modifications can result in either facilitation or competition with other nearby plants, depending on the species and the resources available.

In this thesis, an agent-based model is used to capture the arrangement of plants on a landscape. A fitness function that evaluates both the growth of the individual plants as well as their water use quantifies the value of an arrangement. Using this fitness function and an optimization strategy, the positions of individual plants can be optimized for growth and water use, which offers a new approach to water conservation on residential landscapes.

1.1 What makes this location problem special

Optimization is the process of selecting the best option from a set as defined by a fitness function. In location problems, the options are potential locations for a set of items, and the value of the fitness function represents the relative utility of each location. These problems are often studied as ways to place items to maximize coverage of an area with minimal resources, such as placement of turbines on a windfarm or transmitters in a mobile phone network [71, 106], or to minimize distance traveled between locations or from demand centers [38, 92].

Common to the location problems in the previous paragraph is an explicit representation of distance in the fitness function; changing the distance between items directly affects the fitness of the solution. However, the biological properties of the domain in this thesis generate interactions that make this location problem novel. On landscapes, the distance between plants is relevant only as a measure for how plants affect surrounding resources, which can have a negative or positive effect on nearby plants and their water use. Solutions to this problem are influenced by what I call *neighbor effects*, which include heterogeneity, locality, and feedback. Plants are highly variable, both in their requirements for growth and the effects on their surroundings. Interactions with other plants are primarily local and can have positive and negative effects on growth depending on the types of plants involved and the resources on the landscape. Both heterogeneity and locality

generate feedback conditions, whereby the positioning of a plant on a landscape changes the resources available at that location. The placement of a tree, for example, changes the light available nearby, which then affects the growth for any plants located in those modified conditions. These types of interactions also mean that distance is *implicit* in modeling the effects of the interactions, rather than explicit as it is in most location problems. A shade-loving species will benefit from being close to a shade-generating tree. But, it is the conditions that the tree generates, rather than the explicit distance, that matters. Traditional approaches to solving location problems would not capture these types of interactions.

1.2 An agent-based model for optimizing landscape performance

One approach for finding solutions to problems with complex, local interactions, such as those found in the landscape optimization problem in this thesis, is agent-based modeling (ABM). In these models, individuals in the system are represented as agents with properties and behaviors that allow them to interact with their surroundings and other agents. In the landscape ABM presented here, each plant is an agent with light and water requirements that “lives” on a simulated landscape with light and water properties. Each plant agent’s response to the available resources is described by a growth curve generated from plant growth experiments performed in a greenhouse as part of this research. These growth curves describe how agent growth is influenced by light, water, and competition for water with other agents.

The fitness function for each agent includes two objectives: the growth at the agent’s current location and its water application efficiency (WAE):

$$fit = \alpha \cdot growth + \beta \cdot WAE \quad (1.1)$$

where *growth* is calculated from the biomass produced in the given light and water conditions and *WAE* is calculated from the biomass produced for the amount of water applied. These calculations use the growth curves generated in the greenhouse experiments. The parameters α and β are user-defined weights that address a trade-off in this problem. Growth and water use are conflicting

objectives — plants cannot grow without water. However, reducing water can generate a water savings while still allowing an acceptable level of growth. Modifying α and β enables the user to adjust the priority given to each objective in this multiobjective problem by weighting either *growth* or *WAE* more heavily in the optimization process.

1.2.1 Optimization strategies

Solving an optimization problem requires applying a strategy to maximize (or minimize) the fitness of the solution. In this thesis, the optimization objective is to find the best locations on the landscape for a set of plants, i.e. those maximize fitness of the collection of plant agents:

$$\max \sum_i^n f_i, \quad i \dots n = \text{number of plants} \quad (1.2)$$

The locations in this problem are discrete, making the problem combinatorial. The objective is to select the best k locations from n possibilities, which is an n choose k problem with $\frac{n!}{(n-k)!}$ possible solutions, where k is the number of plants and n is the number of cells on the landscape. For small values of n and k , the optimal solution can be calculated. However, as is the case with most location problems, calculating the optimal solution is computationally intractable for anything beyond simple examples.

Metaheuristics are standard optimization strategies used for finding good solutions to location problems when finding the optimal solution is computationally intractable [100]. Two metaheuristics are evaluated here to determine how they perform on this novel location problem: simulated annealing (SA) [57] and genetic algorithms (GA) [46]. Genetic algorithms are population-based, where a population of candidate solutions is generated and new solutions are in turn generated from these candidates. Solution fitness evolves over several iterations as the fittest members of the population survive to recombine with each other. Simulated annealing starts with a single solution and makes individual adjustments to the solution to improve fitness. On each iteration, a random change is made to the solution. If the change improves fitness, the solution is updated. If the change does not improve fitness, it is still incorporated into the solution with a

probability that decreases as the algorithm progresses. These two metaheuristics are similar in that they evaluate solution fitness globally after each iteration. Computationally, this evaluation entails calculating the fitness for all individuals each time there is a change in the solution.

This thesis presents an optimization strategy designed to capitalize on the novel features of this problem. In this agent-based approach, agents move around on the landscape to maximize their own individual fitness without regard for how their moves affect the global fitness. This method reduces the number of global fitness calculations, as the fitness effect for each move is evaluated for the individual agent only. This reduction in global fitness calculations potentially saves computational time, while still producing solutions that are comparable to the evaluated optimization approaches.

1.3 Experiments and results

Experiments for this thesis include both numerical simulations to evaluate the model under a variety of conditions, as well as validation experiments with live plants to evaluate model predictions about growth and water application efficiency. The numerical simulations evaluated:

- How the three algorithms compared in terms of the fitness scores produced for different scenarios of plant agents.
- The effects of varying the α and β parameters in the fitness function on growth, water application efficiency, and the amount of water applied in the solutions.
- The execution time of the three algorithms.

In the validation experiments with live plants, both a random and an optimized arrangement were generated numerically. In the optimized arrangement, the fitness function weighted WAE more heavily than growth ($\alpha = 0.4, \beta = 0.6$). Plants were grown in the optimized and random configurations for three weeks. Each day, the appropriate amount of water was added to both arrangements (as determined by the model). At the end of the growth period, plants were harvested, dried, and weighed, and the total growth and water use for each arrangement were compared.

The results from these experiments demonstrate the complex interactions that emerge from individual plant placements to determine fitness and water use. Agent fitness in numerical simulations showed the success of modeling interactions through the light and water properties on the landscape. Agent locations in the optimized solutions consistently had light and water properties that matched the agents' light and water requirements. Agents that required shading for survival found these low-light conditions through the optimization process. Competition for water generated spacing between agents that reflected their water requirements and the available water resources. This spacing and the water properties at the individual agent's locations generated the emergent property of total water applied to the landscape.

Complex interactions also showed up in the live-plant validation experiment. The results from this experiment validated model predictions about growth and water use and showed that arrangements can be optimized for water use. While the random arrangement produced more biomass than the optimized arrangement, it did so at the cost of significantly more water. The optimized arrangement produced an acceptable level of growth (all plants appeared healthy throughout the experiment) and used 37% less water than the random arrangement.

There were also interesting results with regards to the optimization strategies. All three strategies produced comparable fitness scores. However, the agent-based search routine ran faster than either SA or the GA. When these three approaches were evaluated on the scores they produced within a given runtime, the agent search produced higher fitness scores than the GA.

1.4 Thesis and contributions

In this thesis, I present the details of a novel location problem and propose that the features of this problem can be captured in an agent-based model.

The contributions of this work include:

- Formulation of a novel location problem that provides a new approach for water conservation on residential landscapes.

- An agent-based model and fitness function to capture and evaluate individual plant growth and efficiency of water use, as well as the emergent total water use from individual behavior.
- Experiments with live plants to calibrate the model and validate claims about optimization, growth, and water use.
- An agent-based search routine that produces comparable results to two metaheuristics in less time.

The remainder of this document proceeds as follows. Chapter 2 presents background on modeling plant growth and resource use on landscapes, which includes single- and multi-species interactions on natural and built systems. Chapter 3 describes location optimization and existing strategies for these problems. Chapter 4 provides background on agent-based models and the types of problems that these models are often used to solve. In Chapter 5, the focus of the document shifts from the background information to the experiments performed in this research. This chapter describes the greenhouse experiments that generated the data for the plant-growth model, which defines agent behavior in different light and water conditions in the agent-based model described in Chapter 6. Chapter 7 describes the agent-based search routine. Chapters 8 and 9 detail the design and results of the numerical simulations and the live-plant validation experiment. Finally, the document concludes with a discussion of the contributions of this work and an outline for future research.

Chapter 2

Modeling landscape resource properties and individual plant growth

Research designed to quantify the costs and benefits of landscaping is generally approached by establishing real landscapes and collecting the relevant statistics over a given period of time [41, 67, 99]. Since testing a landscape configuration can require months or years, which limits the number of studies that can be performed, the way in which a specific multi-species collection of plants will interact on a landscape is often not well understood.

The time scale of live experiments coupled with the seemingly infinite number of spatial arrangements and combinations of different plant species on a landscape motivates the need for computer models to predict landscape performance. In fact, it was stated in 2009,

Methods for designing multi-species systems barely exist...New models are required to represent, assess and design sustainable multi-species cropping systems [65].

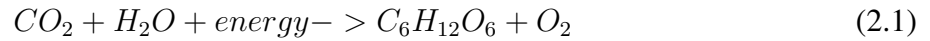
Although this quote referred to multi-species agricultural systems rather than residential landscapes systems, the statement holds true for any built system — there are very few models that capture multi-species interactions.

This chapter covers some of the motivation for, and the current state of, modeling the effects of vegetation on landscapes. This focus starts with a description of models for individual plant performance, including both single-species and multi-species approaches. Many of these models are generated from natural systems and the effects of competition for resources. The multi-species models describe efforts to predict how different species with different resource requirements will

interact on a landscape, and ways to predict water needs in environments characterized by microclimates. This chapter concludes with a review of literature on modeling the interaction of vegetation and resource use in built systems, including species selection for greenroof design, temperature amelioration for shaded patios, and irrigation regimes in a desert city, as further motivation for the problem presented in this thesis.

2.1 Plant water use basics

Plants need water to support the process of photosynthesis:



where carbon dioxide (CO_2) and water (H_2O) combine in the presence of light to produce glucose ($C_6H_{12}O_6$) for the plant and release oxygen (O_2). The process of transpiration provides water for photosynthesis, as plants move water upwards from the soil and eventually lose it out through the leaves in the form of water vapor. Approximately 95% of the water that a plant takes in through its roots is ultimately lost through its leaves [87]. Environmental conditions, such as light, temperature, and humidity all influence a plant's rate of transpiration.

Measuring water loss of a landscape generally includes evaporation and transpiration to create a metric called evapotranspiration (ET). The standard method for calculating ET for a landscape

with a single species is the modified Penman-Monteith (PM) equation, as follows:

$$ET = \frac{\Delta(R_n - G) + \rho_a c_p \frac{e_s - e_a}{r_a}}{\Delta + \gamma(1 + \frac{r_s}{r_a})}$$

where

ET is evapotranspiration in mm/day

R_n is the net radiation

G is the soil heat flux

$(e_s - e_a)$ is the vapor pressure deficit of the air

(2.2)

ρ_a is the mean air density at constant pressure

c_p is the specific heat of the air

Δ is the slope of the saturation vapor pressure temperature relationship

γ is the psychrometric constant

r_s is the (bulk) surface resistance

r_a is the aerodynamic resistance

The aerodynamic resistance, r_a is the transfer of heat and water vapor from the surface into the air above the canopy. This value is calculated as:

$$r_a = \frac{\ln[\frac{z_m - d}{z_{om}}] \ln[\frac{z_h - d}{z_{oh}}]}{k^2 u_z}$$

where

z_m is the height of wind measurements (meters)

z_h is the height of humidity measurements (meters)

d is the zero plane displacement height (meters) (2.3)

z_{om} is the roughness length governing momentum transfer (meters)

z_{oh} is the roughness length governing transfer of heat and vapour (meters)

k is von Karman's constant, 0.41

u_z is wind speed at height z , (m/s)

From this equation, it is clear that there are several environmental variables and plant-related variables that influence water loss. Missing from this equation is the effect of plant *species* on water loss. Species is often included in the form of a crop coefficient, k_c , which is calculated as:

$$k_c = \frac{ET_c}{ET_o} \quad (2.4)$$

where ET_c is the observed ET and ET_o is a reference ET:

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

where

ET_o is reference evapotranspiration in mm/day

R_n is the net radiation at crop surface in $MJ/(m^2 day)$

G is the soil heat flux density in $MJ/(m^2 day)$

T is mean daily air temperature (C) at 2m (2.5)

u_2 is wind speed at 2m height (m/s)

e_s is saturation vapor pressure (kPa)

e_a actual vapor pressure (kPa)

Δ is the slope of the saturation vapor pressure temperature relationship

γ is the psychrometric constant

These equations calculate ET for a well-watered crop. However, the amount of water that a plant transpires is also related to the amount of water in the soil. The transpiration rate can be limited if the water available is not sufficient to support the process. Soils have a maximum amount of water that can be held against the pull of gravity, known as the *field capacity*. When soil is at field capacity, plants can easily extract water from the soil matrix. When the water in the soil drops below a certain value, a plant's ability to extract the water drops exponentially. Studies of this effect have estimated that transpiration declines rapidly when the soil is at 50 to 60% of field capacity [24, 32].

To capture the effects of drought stress in the ET equation, another coefficient, k_s , is often used:

$$ET_c = ET_o * k_c * k_s \quad (2.6)$$

However, even with the crop and the drought stress coefficients, the PM equations are not

really intended to predict water use in a heterogeneous plant population with multiple layers in the plant canopy. Other efforts have focused on predicting water needs using a local evapotranspiration (ET) rate, which is calculated from the environmental conditions and the specific plant species [23]. In that work, a landscape coefficient (k_L), which includes the species, density, and microclimate, replaces the crop and drought stress coefficients, k_c , k_s , to yield:

$$k_L = k_s * k_d * k_{mc} \quad (2.7)$$

where the species factor k_s is a discrete value to represent a species with very low, low, medium, or high water requirements, the density factor k_d is also a discrete value to reflect the planting density — low, average, or high, and the microclimate factor k_{mc} captures environmental variables, such as heat, wind, and shade, to assign a microclimate index of low, average, or high.

In addition to the landscape coefficient method, there are applications that measure the environmental conditions at a particular location and recommend a plant with similar growth requirements [51]. These applications use discrete plant classifications to make recommendations.

The landscape coefficient method is similar to the approach presented in this thesis — plants are divided into categories with discrete requirements and placed into an environment with discrete resources. In this thesis, these resources are light and water; nutrients are not included. This work improves on the landscape coefficient method in its ability to define how microclimates are created and how species will interact on the landscape. For example, planting density contributes to competition for water, light, and nutrients, but this competition also depends on plant size. This level of detail is not included in the landscape coefficient method. Also missing is how growth will be affected under different conditions, i.e. how reducing water will reduce growth. In the next sections, current efforts to address these growth issues are discussed, beginning with methods for modeling plant growth and moving to methods for predicting growth reductions and plant interactions.

2.2 Single-species models

There are several approaches to measuring plant growth. One common approach is to measure the relative growth rate (RGR), the change in plant biomass in grams over time [34]. Another approach is to measure absolute biomass [110]. Both of these approaches have been used to show how plant species respond to different light and water levels [84, 109, 110], and other growth-limiting factors [34]. Water use in plant-growth studies is often represented as grams or liters transpired over a period of time, or as water transpired proportional to the plant's biomass. Another measure, water-use efficiency, measures water transpired as a function of plant growth [24].

Single-species plant models are used to describe the general behavior of a plant species in response to environmental conditions [49]. These models are primarily either empirical or mechanistic in nature. In empirical models, observations are used to describe plant responses to environmental conditions [49]. Examples of empirical models include root growth under a range of environmental conditions [22, 33, 72], light response curves for various species [20, 53, 78], and growth rates and photosynthesis for invasive species [84]. In mechanistic models, the mechanisms of photosynthesis are used to predict how the plant will grow. These models have been used to model plant biomass [94], spatial structure [89], plant growth form [86], crop water use [80], and shade generation from branching structure [86].

In plant ecology, both negative (competition) and positive (facilitation) interactions between plants have been modeled. A type of agent-based model, called *individual based models* (IBM) [50] in this field, are used extensively in ecological research to model competition between plants. Models are also designed to simulate how a single-species population will evolve spatially over time in response to local competition for resources, such as sunlight and rainfall, and other plant-plant interactions [27, 39]. In many of these models, the degree of interaction (competition) between individuals is a function of the distance between plants. Several methods have been used to calculate the expected level of interaction based on a radius around a plant [10, 12, 81, 88, 107, 108]. Other IBMs use a grid to represent the landscape [90], with distance from the plant and inter-

actions based on the cells in the grid. In these models, the size of each cell is selected to represent the space needed for a mature individual of the species.

IBMs have also been combined with cellular automata (CA), a type of agent-based model where the agents have discrete states and exist on a discrete landscape [79]. One such study modeled competition for light for two theoretical species [40]. Each species was given a rank that reflected its ability to capture resources and to disperse seeds. The study found that the species with greater resource capture was more competitive than the species with greater dispersal abilities and thus spread more quickly. In another study, a vegetative growth model was based on biomass and soil moisture, and a CA was used to describe how the vegetation spread [2].

2.3 Multi-species models

The previous section described models of single-species growth and interactions. There are also multi-species models that capture how some species compete for resources, while others can have a relationship that is beneficial for at least one of the species. In one class of competitive models, known as *ecological field theory* (EFT), competition is based on spatial interactions and resource use. In EFT models, each plant's influence on light, water, and nutrients is used to calculate an interference potential around the plant [95, 105, 114], which determines seedling establishment and growth of larger plants.

These positive relationships — known as facilitation — often show up in extreme environments where one species provides protection for another [17]. Both facilitation and competition have been included in models for predicting seed germination and the effects of resource constraints on growth [62, 114]. Ecological field theory and statistical mechanics have been used to show how interactions between individuals leads to self-thinning [62]. This general model defines the effects of competition and facilitation between individual plants by including positive and negative weights for determining a plant's zone of influence.

Models of plant facilitation are less common than competition models, but have been used to explore interactions in many types of ecosystems [15]. One study of interactions between shrubs

and annuals found that the degree of facilitation was related to aridity on the landscape—as arid conditions increased, so did facilitation [47, 101]. Another study produced a similar result—showing that facilitation increased as abiotic stress increased in alpine environments [18]. There have also been theoretical simulations to examine the roles of positive and negative interactions between plant species [102]. Plant interactions have also been examined as a tool for landscape rehabilitation to determine whether existing vegetation helps or hinders the establishment of new species [37].

2.4 The need for multi-species optimization

All existing research on modeling plant growth, plant interactions, and water use requirements is aimed at predicting behavior under a set of conditions. As stated at the beginning of this chapter, there is a need for multi-species models to capture both facilitation and competition. More than that, there is an opportunity to use these models to optimize the value that vegetation provides in urban environments. Studies of multi-species interactions in these environments have shown how species composition affected green-roof design [64], cooling [96], and microclimate generation [98]. However, none of these studies optimized these species interactions, or the placement of the individual species.

In one research study that used computational methods to find good locations for plant species, GIS-based methods were used to locate potential tree planting sites in Los Angeles, California [113]. The selection criteria for potential locations included enough room to support growth and distance from other trees and structures. However, this study did not include interactions or even multiple tree species with different growth requirements.

In addition to the need for placement optimization, there is a need for optimization of water application. Several studies have shown that water application practices on residential landscapes have room for improvement. Some studies have shown that many landscape plants perform well receiving only 50% of the reference ET_o [41]. Another study examining water application rates of xeric and mesic landscapes in a desert city found that irrigated landscapes use water less efficiently

than the nearby native desert, and that water use efficiency on residential landscapes could be improved by reducing irrigation [67]. Other work by the same authors found that homeowners frequently over-watered their landscapes, even those with desert species, effectively eliminating the benefit of having a drought-tolerant landscape [66].

In the next chapter, the concept of a location problem is presented, including both what a location problem is and how these problems are generally solved. This background information provides the foundation for discussing landscape design as a location problem in which the objective is to find the best places of a residential landscape for a collection of plants.

Chapter 3

Location optimization

Optimization is the process of choosing the best member of a set of options, where the value of each option is defined by a fitness function [82]. This problem can be defined formally by the tuple (S, f) , where S is the state space and f is the objective function that assigns a value to every $s \in S$. The optimal solution is the one that minimizes or maximizes the fitness function, such that $\forall s \in S, f(s^*) \leq f(s)$ [100].

Location problems are a class of optimization problems where the state space S is a set of physical locations and the objective is to find the best locations for a set of items using the value assigned to each location $s \in S$ by the objective function f [42]. In these problems, the candidate locations can be continuous — providing an infinite numbers of solutions. The locations can also be discrete, which creates a combinatorial problem with a finite set of candidate solutions [82].

This chapter provides the background information on combinatorial location optimization that is needed to understand the problem described in this thesis. The information presented includes:

- The history of location problems and the types of problems currently studied.
- An overview of other combinatorial problems.
- A description of the special features of the landscape optimization problem presented in this thesis.
- An overview of optimization algorithms.

3.1 Location problems

Examples of location problems can be found across multiple subject areas. One of the most common is that of facility location, also known as the p-median problem or the quadratic assignment problem, where the objective is to determine the locations of n facilities, selected from m locations, ($m > n$), that minimize the distance traveled between facilities with the highest flow [92]. In one common formulation of this problem, the objective function is:

$$\min \sum_{j=1}^n \sum_{i=1}^m a_i d_{i,j} x_{i,j}$$

where

(3.1)

a_i is the population of community i , $i = 1, 2, \dots, n$

$d_{i,j}$ is the distance between community i and facility j

$$x_{i,j} = \begin{cases} 1 & \text{community } i \text{ assigned to facility } j \\ 0 & \text{otherwise} \end{cases}$$
(3.2)

In this location problem, distance traveled is a significant component in the optimization criteria (see [58] for a review). In other optimization problems, such as max-cover, the objective is to maximize coverage over an area. This approach has been used in several real-world location problems, such as locating emergency medical services [29], warning sirens [25], water monitoring stations [60], rain gauges [45], and mobile phone transmitters [106].

In location problems, the position of an item can create competition with other items on the landscape. This phenomenon was seen in a transmitter study [106], where the position of a transmitter generated competition for the signal with other transmitters. Competition was modeled by including an explicit distance between the transmitters in the fitness function, i.e. similar transmitter produced less competition if they were farther apart. Another study that examined competition for clients by retail establishments also used distance in the fitness function to control placements [14].

Other studies of location problems addressed competition in another way. In a minimum-set cover problem involving wind farm design [71], the objective was to place turbines in such a way

as to maximize energy production, while minimizing the number of turbines and acreage needed. The wind patterns were variable over the landscape and the placement of one turbine affected the surrounding wind patterns, which in turn affected the placement of the next turbine. To address this interaction issue, the landscape resolution was adjusted — the size of grid cells was set such that a turbine in a grid cell did not affect the wind patterns in any other cell. Although in this case, distance was not explicit in the objective function, it was still an explicit part of the optimization problem, as it affected the minimum spacing between the turbines.

Another element to consider in location problems is whether the items are all the same. In the examples presented above, the wind turbines all had the same energy-producing capacity, the facilities could all handle the same demand, and the transmitters all had the same signal strength. However, in [38], a location problem for health centers was discussed in which each health center had a different medical specialty. The objective was to determine the location, size, and services for each facility that minimized the distance traveled from population centers to the health care facilities and to minimize overlap in coverage at the facilities.

3.2 Multiobjective optimization

In many optimization problems, there can be multiple, conflicting criteria. These problems, known as multiobjective optimization problems (MOP) are defined as:

$$MOP = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ s.c. x \in S \end{cases} \quad (3.3)$$

where $(n \geq 2)$ is the number of objectives, $x = (x_1, x_2, \dots, x_k)$ are the decision variables, and S is the set of feasible solutions [100]. An example of a multiobjective optimization problem is the wind farm study in [71]. There, the objectives were to produce the highest amount of energy at the minimum cost. These conflicting objectives were captured in the fitness function as:

$$f = \frac{1}{P_{tot}}w_1 + \frac{cost_{tot}}{P_{tot}}w_2 \quad (3.4)$$

where P_{tot} is the total energy, $cost_{tot}$ is the total cost, and w_1 and w_2 are arbitrarily chosen weights. The values selected for w_1 and w_2 determine the significance of each optimization criterion in the final solution — i.e. which one of P_{tot} and $cost_{tot}$ is more important.

The example in the previous paragraph introduces the concept of how multiobjective problems can often have many solutions that cover a range of the optimization criteria. However, even when there are multiple solutions, some are better than others. These “better” solutions, which are called *Pareto optimal*, are defined formally as:

A solution $x^ \in S$ is Pareto optimal if for every $x \in S$, $F(x)$ does not dominate $F(x^*)$, that is $F(x) \not\prec F(x^*)$. [100]*

The main goal of multiobjective optimization is to find the Pareto-optimal set of solutions for a problem, which are said to make up the Pareto front. Solutions that are worse than those on the Pareto front are said to be *dominated*. For example, assume in (3.4) that $P_{tot} = 200$ and $cost_{tot} = 100$. Solutions where $P_{tot} = 200$ and $cost_{tot} > 100$ would be dominated, since there was not a corresponding increase in P_{tot} for the increase in $cost_{tot}$.

One of the challenges of multiobjective optimization is combining the fitness objectives into a single fitness function. A common method for this is assigning a weight to each objective, which requires determining the correct weights for each objective. Another common approach is to optimize one objective subject to the constraints of the other objectives. This is written as:

$$MOP_k(\epsilon) = \begin{cases} \min f_k(x) \\ x \in S \\ s.t. f_j(x) \leq \epsilon_j \end{cases} \quad (3.5)$$

where, using the previous example in Equation (3.4), $f_k(x) = \frac{1}{P_{tot}}$, $f_j(x) = cost_{tot}$, and ϵ_j is a fixed cost constraint. This approach essentially converts some of the optimization objectives to constraints [100].

The landscape optimization problem in this thesis contains multiple potentially conflicting objectives: to maximize the growth of the individual plants and to minimize water use on the landscape. As such, the problem is a multiobjective optimization problem. Defining the fitness

function in a similar manner to the examples above provides a few options. One possible definition is:

$$f = bio_{tot} \cdot w_1 + WAE \cdot w_2 \quad (3.6)$$

where bio_{tot} is the total biomass on the landscape and WAE is the water application efficiency: $\frac{bio_{tot}}{water_{tot}}$, where $water_{tot}$ is the total water used.

Another option for a fitness function for landscape optimization is:

$$MOP_k(\epsilon) = \begin{cases} \max f_k(x) \\ x \in S \\ s.t. f_j(x) \leq \epsilon_j \end{cases} \quad (3.7)$$

where $f_k(x)$ is biomass produced, $f_j(x)$ is water used, and ϵ_j is a fixed water budget for the landscape. In a real-world situation, this water budget could be set by how frequently a person wanted to water and how much water the irrigation system delivered with each watering.

3.3 Other combinatorial optimization problems

In addition to combinatorial location problems, there are other combinatorial problems frequently studied in computer science that are relevant to the landscape optimization problem presented in this thesis. Two relevant instances are the traveling salesman problem and the knapsack problem. In the traveling salesman problem, the objective is to find the shortest path through a set of nodes (cities) without visiting the same node twice. Each solution consists of a set of nodes in the order that they are visited, together with the corresponding distance. Distance is again explicit in the fitness function. However, the focus is not on placement of objects but rather on travel between objects. In the knapsack problem, the objective is to select a subset of objects to maximize profit for a fixed weight:

$$\begin{aligned} \max \quad & \sum_{i=1}^n v_i x_i \\ \text{s.t.} \quad & \\ & \sum_{i=1}^n w_i x_i \leq W \end{aligned} \quad (3.8)$$

where v_i is the value of object i , w_i is the weight of object i , and W is the weight constraint. In this problem, distance is clearly not part of the fitness function; the issue is selecting the best subset of items. Once an item is selected, it changes the space remaining for future items. This feature is similar to the way placing a plant in the landscape problem changes the surrounding light and water resources for additional plants.

3.4 A different kind of location problem

In choosing locations for plants on a grid, the objective is to select the best k out of n possible locations, where n is the number of cells that make up the landscape and k is the given number of plants. This task has a complexity of $\frac{n!}{(n-k)!}$, and is similar to the combinatorial location problems discussed above. The objective is:

$$\min f_k(x), \forall k \quad (3.9)$$

Plants on the landscape interact with surrounding plants by changing light resources and competing for water. These interactions are similar to the flow between facilities, or between facilities and demand centers, or a wind turbine affecting the wind that nearby turbines receive. Selecting the best subset of locations is also similar to the task of selecting a subset of items in the knapsack problem. Each selected location contributes to an overall value of the arrangement, while incurring a cost — water. On a real landscape, there could be other associated costs, such as the cost of the individual plants. However, since the problem presented in this thesis is about water conservation, water is the only cost being addressed.

There are complexities in selecting plant locations on a landscape that are not present in either the location problems or the knapsack problem. One difference lies in how distance between items contributes to the fitness function. In most location problems, this distance is an explicit part of the optimization criterion[31]. On a landscape (both real and simulated), plants' influence on their surroundings is based on distance, but distance is not explicit in the fitness function. That is, *the conditions generated by the distance* are important, but not the distance itself. This effect

is illustrated in Figure 3.1. In this image, a large tree is shading surrounding cells. The effect of this shading depends on the plants' requirements, rather than the distance to the tree generating the shade.

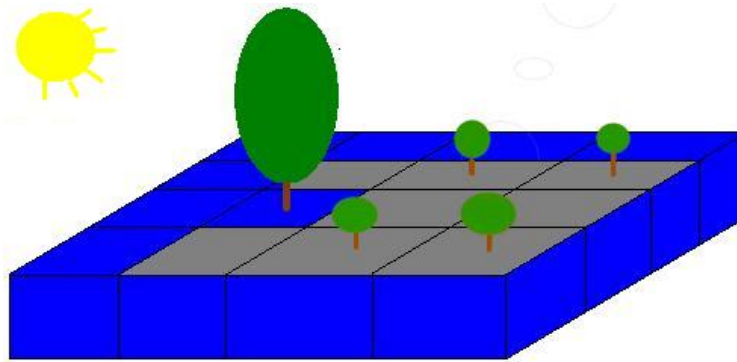


Figure 3.1: The position of one plant can affect other plants by modifying the light at each location. In this example, the tree is generating shade in surrounding cells, which changes the light available for the other plants. The shaded cells are shown in grey.

3.4.1 Neighbor effects

Approaching the landscape optimization problem by focusing only on the distances between plants would not capture the complex interactions in this problem, and thus would not likely produce a good solution. In this dissertation, I term these interactions *neighbor effects*, and define them as locality, heterogeneity, and feedback. Plants are heterogeneous, both in their growth requirements and their effects on their surroundings. Interactions with other plants are primarily local, and can either increase or decrease growth depending on the types of plants involved and the resources on the landscape. These types of interactions also mean that distance is *implicit* in modeling the effects of the interactions, rather than *explicit* as it is in most location problems.

Both heterogeneity and locality generate feedback conditions, whereby the positioning of a plant on a landscape changes the resources available at that location. The placement of a tree, for example, changes the light available nearby, which then affects the growth for any plants located in those modified conditions. These elements of locality and feedback are missing from traditional

location problems. For example, placing a facility to meet demand does not change the demand around the facility, it just satisfies the demand. However, placing a tree changes the sunlight—which changes the types of plants that grow well near the tree.

3.5 Optimization algorithms

There are several traditional methods for solving combinatorial optimization problems. For smaller problems, branch and bound algorithms are common [100]. In these algorithms, a tree is constructed of all possible solutions and branches that do not contain any optimal solutions are pruned dynamically. Another common approach for small problems is linear programming (LP), which is formulated as:

$$\begin{aligned}
 & \min cx \\
 & s.t. \\
 & Ax \geq b \\
 & x \geq 0
 \end{aligned} \tag{3.10}$$

LP can be used to find the optimal solution for problems where the objective function and the constraints are linear functions. However, real-world optimization problems are often complex and do not fit neatly into an LP approach. For these types of problems, finding the optimal solution is computationally intractable, and the focus instead is generally on finding a “good” solution, where the definition of “good” can vary depending on the particular problem. Approximation algorithms and heuristic methods are common for this task. The former find solutions that are guaranteed to be within some bound of the optimal solution [43]. In contrast, heuristic methods find good solutions in a reasonable time, but there is no guarantee about how close the solution is to optimal. Within the class of heuristic methods, some are designed to solve a specific problem, while others, called *metaheuristics*, are general-purpose algorithms. These general-purpose metaheuristics are the focus of this section.

3.5.1 Metaheuristics

Several types of metaheuristics have been used to solve location optimization problems. These algorithms can be divided into two main classes — single-solution or population-based methods. In single solution methods, such as local search [1], simulated annealing [57], and TABU search [36], a single solution is iteratively improved using a set of rules to explore the search space. In population-based methods, such as genetic algorithms [46], scatter search [35], and particle swarm optimization [56], a population of solutions is generated randomly and new solutions are created through combinations of this population. The following section provides a more-detailed description of the metaheuristics that are relevant to the landscape optimization problem in this thesis.

3.5.1.1 Local search

One of the oldest and simplest heuristics is local search [1]. This algorithm generally begins with a randomly generated solution, then iteratively improves the current solution by making small adjustments that increase solution fitness. A template of the algorithm is provided in Algorithm 1. On each iteration, the candidate solution that most improves the current solution is selected as the new solution. The search stops when all new candidate solutions are worse than the current solution. Local search is prone to getting stuck in local optima, as it does not provide a method for selecting candidate solutions outside of a local neighborhood. This algorithm is also very sensitive to the quality of the initial solution. Several algorithms have been developed that include local search as one component of another metaheuristic, e.g. [52, 75, 103].

3.5.1.2 Genetic algorithms

Genetic algorithms (GA) are designed to mimic the process of evolution. The GA begins with a population of solutions, generated randomly. Then, using computational mechanisms that model the processes observed in natural systems — inheritance, natural selection, and mutation —

Algorithm 1 Local search algorithm

```

1: Generate an initial solution,  $S_0$ .
2: repeat
3:   Generate neighbor candidate solutions.
4:   Select best candidate solution,  $S_c$ .
5:   if  $S_c > S_0$  then
6:      $S_0 = S_c$ 
7:   end if
8: until No candidate solutions improve  $S_0$ 
9: Return  $S_0$ 

```

the population evolves as individual solutions with high fitness scores survive, while low fitness solutions die off. Each candidate solution is called a *chromosome*, and each element on a chromosome is an *allele* that contains a value used in the fitness calculation. Typically, a chromosome is represented as a string of 1s and 0s, where each position in the string represents an item in the solution. A 1 means to include that item and a 0 means to leave that item out of the solution. For example, in the knapsack problem, the objective is to find the subset of items that maximize value in a fixed space. The chromosome of 1s and 0s represents the potential items to add to the knapsack. A chromosome of 10101010, for example, is a solution that selects the first, third, fifth, and seventh items. The value and size of these items is included in the fitness calculation to generate a fitness score for that chromosome. The population of solutions is a set of strings with different combinations of 1s and 0s.

An example GA is shown in Algorithm 2. Starting from a population of a randomly generated solutions, the algorithm selects two members and generates an offspring. This process, known as *crossover*, takes a portion of the solution string from each parent and combines them to generate a new solution. Selecting the crossover point and determining which part of the strings to select from each parent are governed by parameters of the GA. For example, starting with the parents 101010 and 000111, and a crossover point in the middle of the solution, the child solutions would be 101111 or 000010. A child solution is added to the population if it improves the overall fitness of the population. Another step in the GA is *mutation*, which is designed to prevent the algorithm from getting stuck in local optima. In the mutation step, some number of alleles are

selected randomly and their values are changed. For example, if the solution 000000 is mutated at the third allele, the new solution is 001000.

Algorithm 2 Genetic algorithm

- 1: Set *PopSize*, the number of individuals in the population.
 - 2: Set *MutRate*, the rate of mutation in the population.
 - 3: Generate population of individuals of 0-1 strings, called *Pop*.
 - 4: **repeat**
 - 5: Sort *Pop* based on solution fitness.
 - 6: Randomly select two members from *Pop*.
 - 7: Generate new offspring using crossover on two members.
 - 8: Insert offspring at end of *Pop* and re-sort.
 - 9: Truncate *Pop* to the *PopSize* fittest individuals.
 - 10: Randomly select individuals for mutation, using *MutRate*, and flip 0-1.
 - 11: Re-sort *Pop*.
 - 12: **until** fixed number of iterations reached
-

An important step in producing a GA is deciding what each allele represents in the solution. Several approaches have been used in applications of GAs to location problems. Alleles can represent potential locations, and the GA can be used to select the subset of locations that maximize fitness, e.g. [14, 26, 48, 71]. This approach works well for max-cover problems where the objective is to maximize coverage of an area for the minimum number of nodes. For example, in [71], the objective was to minimize the number of wind turbines needed for the maximum energy production. The 2D landscape of $m \times n$ cells in which turbines could be sited was flattened to a 1D representation of $m * n$ alleles; a 1 in the solution meant that a turbine was placed at that position and a 0 meant no turbine at that position. The GA determined both the number of turbines and their positions on the landscape.

In other facility location problems, alleles were encoded to represent facility indices [3, 28, 70]; here the number of items in the chromosome was the number of facilities to place. For example, if locations for four facilities were needed, one possible encoding would be [2, 4, 6, 8], meaning “select the locations 2, 4, 6, and 8 for the facilities”. This approach is different from the wind turbine example in the previous paragraph — it controls the number of items being placed by using the alleles to represent *items* instead of locations. This approach does introduce

some difficulties. For example, if the chromosomes $[1, 2, 3, 4]$ and $[3, 4, 5, 6]$ are crossed to get $[3, 4, 3, 4]$, locations 3 and 4 are both selected for two facilities. Different repair methods have been proposed to address this issue. One approach is to combine the solutions, rather than use a crossover method, which in this example would produce $[1, 2, 3, 4, 5, 6]$. The two least-fit items are then removed from the solution. Another approach is to perform the traditional crossover and then apply a penalty, or to remove non-unique items from the solution by replacing them with other randomly selected items, such as discussed in [48, 106].

Although GAs have been used for location problems, the neighbor effects that occur in the landscape problem make it difficult to know how GAs will perform on the problem described in this thesis. There are two key reasons for this:

The 0-1 encoding is not sufficient In most GAs, the solution is encoded as a binary string, as described above, with 1s and 0s used to indicate the presence or absence of an item at a particular location. However, in the landscape problem, both the location and the *type of plant at that location* can change. The encoding needs to capture this additional detail, which is impossible with only two options.

Local interactions and feedback When the alleles on the GA chromosome represent locations, those alleles are independent, i.e. the selection of one allele does not affect the properties of another. However, in the landscape problem, locality and feedback can generate positive interactions between individuals that are in close proximity. The traditional method of crossover, then, may not preserve the interactions that generate a good fitness score for the chromosome. For example, consider the example shown in Figure 3.1 of a 4x4 grid occupied by a tree and four smaller plants. This example can be converted into a matrix of plant placements, shown in Table 3.1, where the locations labeled 1 contain a plant and a 0 is a location with no plant.¹

Generating a 1D chromosome from this 2D example produces:

¹ This matrix is a simplified example of the placements in Figure 3.1 in that it does not capture the heterogeneity of the species. However, it is used here to show feedback onto the landscape

Table 3.1: Plant placement matrix, where 1 indicates the placement of a plant and 0 is no plant.

0	0	0	0
0	0	1	1
0	1	0	0
0	0	1	1

$$[0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1]. \quad (3.11)$$

Breaking the chromosome in the middle produces two pieces:

$$[0, 0, 0, 0, 0, 0, 0, 1, 1] \text{ and } [0, 1, 0, 0, 0, 0, 0, 1, 1] \quad (3.12)$$

that are used in crossovers with other chromosomes. This process could change the light conditions for two plants, since the plant that is generating shade is on the other half of the chromosome. This issue would also arise if the chromosome were comprised of individual plants. The crossover process could still separate plants from the conditions they need for good fitness.

A landscape GA Appropriate GA encoding is a design decision in any optimization problem. In this thesis, the encoding is similar to that used in [3, 28, 70], where each allele on the chromosome was a facility with a location index. Here, the alleles are plants with a discrete location. For example, the Landscape image in Table 3.2 shows a 4x4 landscape with six plants, each with a unique ID to signify the type of plant, and a discrete x,y location. In the GA chromosome created from this landscape, shown in the GA Chromosome image in the same table, each allele on the chromosome is a plant with a set of properties and a discrete set of x,y coordinates that describe its location. To generate new solutions, two chromosomes are crossed at the midpoint, as shown in the GA Crossover image, such that each new solution contains half of the parents' properties. Mutations in this encoding are changes to the position of individual plants, also shown in Table 3.2.

Using this encoding, the GA proceeds as shown in Algorithm 3. Most of the steps shown here are standard for GAs, as shown in Algorithm 2. However, there are a few noteworthy differences. In line 3 of Algorithm 3, the population is defined as a set of chromosomes where each

Table 3.2: 4x4 landscape with six plants converted to a GA Chromosome. The GA Crossover and GA Mutation images here also show the process of crossover and mutation, which are how new solutions are created from the existing population.

Landscape

P1 $\{x, y\} = 0,0$			
		P3 $\{x, y\} = 1,2$	P5 $\{x, y\} = 1,3$
	P2 $\{x, y\} = 2,1$		
		P4 $\{x, y\} = 3,2$	P6 $\{x, y\} = 3,3$

GA Chromosome

P1 $\{x, y\} = 0,0$	P2 $\{x, y\} = 2,1$	P3 $\{x, y\} = 1,2$	P4 $\{x, y\} = 3,2$	P5 $\{x, y\} = 1,3$	P6 $\{x, y\} = 3,3$
------------------------	------------------------	------------------------	------------------------	------------------------	------------------------

GA Crossover

P1 $\{x, y\} = 0,0$	P2 $\{x, y\} = 2,1$	P3 $\{x, y\} = 1,2$	P4 $\{x, y\} = 3,2$	P5 $\{x, y\} = 1,3$	P6 $\{x, y\} = 3,3$
P1 $\{x, y\} = 1,0$	P2 $\{x, y\} = 2,2$	P3 $\{x, y\} = 0,2$	P4 $\{x, y\} = 3,3$	P5 $\{x, y\} = 1,2$	P6 $\{x, y\} = 3,0$



P1 $\{x, y\} = 0,0$	P2 $\{x, y\} = 2,1$	P3 $\{x, y\} = 1,2$	P4 $\{x, y\} = 3,3$	P5 $\{x, y\} = 1,2$	P6 $\{x, y\} = 3,0$
P1 $\{x, y\} = 1,0$	P2 $\{x, y\} = 2,2$	P3 $\{x, y\} = 0,2$	P4 $\{x, y\} = 3,2$	P5 $\{x, y\} = 1,3$	P6 $\{x, y\} = 3,3$

GA Mutation

P5 $\{x, y\} = 2,0$

allele is the same plant, but with different locations. To initialize the population, random locations are generated for all plants. In line 8, the algorithm checks that all locations are unique on a chromosome following crossover. One of the limitations of this GA encoding is that crossover and mutation can create the condition where multiple plants are at the same location. When this happens, a repair step moves one of the plants to another location (line 9). The same process is used following mutation, which selects a new random location for a plant. If this process generates a location conflict, another random location is selected.

Algorithm 3 Genetic algorithm for landscape optimization

- 1: Set *PopSize*, the number of individuals in the population.
 - 2: Set *MutRate*, the rate of mutation in the population.
 - 3: Generate population of individuals with random locations, called *Pop*.
 - 4: **repeat**
 - 5: Sort *Pop* based on solution fitness.
 - 6: Randomly select two members from *Pop*.
 - 7: Generate new offspring using crossover on two members.
 - 8: **if** Location for each element in offspring is not unique **then**
 - 9: Generate new random location for one element.
 - 10: **end if**
 - 11: Insert offspring at end of *Pop* and re-sort.
 - 12: Truncate *Pop* to the *PopSize* fittest individuals.
 - 13: Randomly select individuals for mutation, using *MutRate*.
 - 14: Generate new random location for mutated individuals.
 - 15: **if** Location for each element in offspring is not unique **then**
 - 16: Generate new random location for mutated element.
 - 17: **end if**
 - 18: Re-sort *Pop*.
 - 19: **until** fixed number of iterations reached
-

3.5.1.3 Simulated annealing

Simulated annealing (SA) is a metaheuristic that uses a very different approach than GAs. Instead of evolving fitness scores using a population of solutions, SA starts with one solution, and makes incremental improvements by modifying individual items. This algorithm is designed to mimic the process of annealing metals [77]. In a high-energy state, particles move around rapidly within the metal; as it cools, particles slow down. SA emulates this using a temperature parameter

that is set to a high value at the beginning and slowly cooled over the run. When the temperature is high, the algorithm accepts moves that do not improve solution fitness. As the optimization runs and the temperature cools, the probability of accepting a bad move gets lower. A template of an SA algorithm is provided in Algorithm 4. In this algorithm, an initial random solution is generated and the fitness score for that solution is calculated. For a location problem, this initial solution is a random location for each item. Next, a new random location is selected for each item, one at a time, and the new fitness score is calculated. If that single move has improved the global fitness, the item is moved to that location. Otherwise, with some decreasing probability, the item is moved regardless of its affect on the global fitness to keep the search from getting stuck in local optima.

Algorithm 4 Simulated annealing

```

1: Set starting temperature,  $T = T_{max}$ 
2: Generate initial solution,  $S = S_0$ 
3: repeat
4:   for all  $s \in S$  do
5:     Generate random neighbor solution  $S'$ 
6:     Calculate fitness  $f_{S'}$ 
7:      $\Delta E = f_{S'} - f_S$ 
8:     if  $\Delta E \leq 0$  then
9:        $S = S'$ 
10:    else
11:      Accept with probability  $e^{\frac{-\Delta E}{T}}$ 
12:    end if
13:  end for
14:   $T = \alpha T$ 
15: until  $T < \epsilon$ 

```

SA is fundamentally different from GAs in that each iteration involves a change to an individual element in the solution instead of a blending of multiple elements from multiple solutions. These two methods span the space of existing methods, and therefore, both are being evaluated here. Although SA has been used to solve a variety of location problems [21, 76, 77], just as with the GA, it is unknown how well SA will capture the neighbor effects in this problem. The global fitness evaluation after each individual move may mean that dependent elements stay together in SA iterations, as individual moves that result in decreased fitness for surrounding elements are

identified by the reduced global fitness score. However, moves in SA are also often selected randomly, which may not capture the small adjustments needed to maximize fitness.

3.5.2 Agent-based methods

Agent-based optimization routines are not common in location problems. However, agent-based models (ABM) are often used to explore systems where behavior is influenced by neighbor effects — locality, heterogeneity, and feedback. Therefore, it is worth exploring their value in the context of this problem. In metaheuristics, the fitness of every individual item in the system is included in calculating the value of an optimization step. In agent-based systems, each individual maximizes its own fitness by moving around on the landscape in response to local conditions. In this way, system fitness “emerges” from the fitness of the individuals. (Agent-based models are discussed in more detail in the next chapter.)

Although not common, there are examples in the literature of applying agent-based optimization routines to location problems [4]. In one such study, agent-based models were used to locate bus stops [74]. Space was defined as continuous (instead of a discrete grid as in the landscape problem), and agents reacted to potential fields surrounding each location and optimized their own position; the global fitness followed from the ensemble of those local conditions. Another facility location study used a similar approach of attractive and repulsive forces, where agents were attracted to demand centers and repelled from other agents [73]. In this study, just as in the location problems discussed earlier in the chapter, distance had explicit effects on determining agent fitness — agents gained fitness as they moved toward demand centers and away from other agents.

Another approach to finding good locations for items in multi-agent systems exists in situated multi-agent systems [8, 30]. Again, these models use distance between agents to determine fitness, an approach that does not capture the positive and negative influence between individuals that plays such a critical role in the landscape problem presented in this thesis. These models have been used to locate shopping centers using reaction fields similar to the approach discussed in [73].

In another agent-based optimization approach, a multi-agent algorithm was used to solve

constraint satisfaction problems [63]. This approach was applied to two NP-complete problems, the n -queen problem and the graph-coloring problem, where each item was an agent with behaviors that allowed it to react to local surroundings. While not a location optimization problem, this algorithm did feature agents reacting to their surroundings to optimize their individual positions. There were differences between this algorithm and the agent-based method presented in this thesis. The problems studied were simpler than the landscape optimization problem studied here. The items were uniform and the agents moved on a 1D grid. Significantly, the position of an agent did not change the conditions in nearby grid cells.

This chapter presented the fundamentals of location optimization and showed how the landscape optimization problem is different than traditionally studied problems. This difference is primarily related to measuring the explicit distance between items in the fitness function. In the landscape problem, plants affect their surroundings in ways that are positive for some other plants and negative for others. Therefore, it is not distance between plants that matters, but rather the effects that plants have on their surroundings and how other plants are then affected by these modified conditions. One class of methods for modeling these types of interactions are agent-based models, which are discussed in the next chapter.

Chapter 4

Agent-based modeling

Agent-based models (ABM) are a class of computational models that simulate the behavior of autonomous individuals. In these systems, each individual is represented as an agent, implemented as a software object, with a set of properties and behaviors that control its movements and its interactions with surrounding agents and its environment [112]. This ability to model individual actions means that a heterogeneous agent population can be created by simply assigning different properties and behaviors to individual agents. Understanding the complex system-level behavior that emerges from individual agent actions and interactions is accomplished through simulation, which involves running the model for a given set of inputs.

While the implementation of individual agents may be simple, the types of problems that ABMs are used to solve are often quite complicated. These models are frequently applied to problems that cannot be solved analytically, in particular problems where spatial interactions are a key element in the solution. Giving agents a physical location can be accomplished simply by making location an agent property, and updating this value effectively moves the agent to a different location. If an agent's location is known, then determining which agents are nearby and implementing interactions based on this proximity is a trivial process. The types of interactions that occur between agents can be instantiated as methods in the agent object. ABMs are also useful when some equations describing a system are known, but the system also includes stochastic processes that make it difficult to define behavior.

For the reasons listed above — heterogeneity, spatial interaction, local influence — ABMs

are a natural fit for describing plant interactions on a landscape. Each plant can be instantiated as an agent with a set of properties and a spatial location. Each plant agent's effects on surrounding resources and agents emerge naturally from its location, together with individual behavioral rules for growth and water use. This approach is good not only for modeling plant agent behavior, but also for optimizing the position of individual plant agents.

There are several decisions that need to be made when one constructs an agent-based model [112], including:

- What are the properties and behaviors of the agents?
- How will the agents be embedded in the environment, and how will they interact with it and each other?
- What are the appropriate scales for time and space to capture the relevant dynamics in a given simulation?
- Will agents “learn” from their interactions?
- How will the model be used?
- How will the model be validated?

The body of literature about where and how ABMs have been used is extensive and an exhaustive treatment is beyond the scope of this thesis. The remainder of this chapter describes examples of ABMs that are relevant to the decision criteria listed above, as well as one of the current limitations of ABMs, validation. Most of these examples include models of heterogeneous individuals with explicit spatial properties and local interactions. Some are used for optimization, while others are used for decision support. (Plant arrangements are presented here as an optimization problem, but it is not difficult to see how presenting water-use behavior and arrangements could be viewed as a decision support tool.) Finally, methods for validation are presented. ABMs are frequently used to evaluate systems that change over large spatio-temporal scales and include non-linear interactions. Reproducing these interactions on measureable scales can be difficult, if not impossible.

4.1 Determining properties and behaviors

One of the earliest ABMs was BOIDS [93]. In this model, each agent was a bird that behaved following three simple rules:

- To fly towards the center of mass of neighboring boids.
- To keep a small distance away from other objects (including other boids).
- To match velocity with nearby boids.

The behavior of the agents that emerged from these interactions visually resembled the flocking behavior of real birds, even with no central control guiding the emergent flock behavior. In the BOIDS model, the agents were homogeneous.

Many of the models that have been created since then include heterogeneous agents with more complex interactions. The content of these models is not limited by discipline or problem type. For example, in a simple ABM in which each agent is a trader in a stock market, agents could have the following implementation:

Agent trader: An individual trader in the stock market.

- Properties: The trader's features that motivate his actions in different situations.
 - * Assets: How much money the trader has at a particular time.
 - * Risk: How likely the trader is to buy a high-risk, potentially high-reward stock.
- Methods: The trader's actions in a situation.
 - * Buy stock using the trader's assets.
 - * Sell stock to receive assets.

In this example, the properties are assets — the amount the trader has to work with in trading, and risk — the trader’s likelihood of buying or selling a high-risk stock. The agent’s actions are buying and selling, and the result of each action updates the agent’s assets.

ABMs have also been used to model biological systems such as intracellular reaction rates [85] and cell growth [55, 104]. In the reaction-rates model, each reaction element was represented as an agent that reacted with other nearby agents following established rules for reactions. In the cell-growth model in [55], agents were the different components of a cell. Resources were differentially allocated to agents to model the cell’s ability to consume different types of external substrate, which determines cell growth. In another cellular-interaction model of epithelial cells [104], each agent modeled a cell that had a physical location in an environment and a size. Cells grew and moved; they interacted with neighboring cells by bonding or repelling. The time scale for these simulations was based on laboratory experiments on real cellular interactions, i.e. the ABMs included equations for observed behavior, and produced results that were qualitatively similar to observed reaction rates.

4.2 Agent interactions

The task of determining how simulated agents will exist in the environment and will interact with other agents includes defining space and the complexity of agent interactions. When agents are placed in a simulated version of a physical environment, space can be discrete, as in [55], or continuous, as in [74]. The spatial interactions in the environment can be an explicit function of distance between agents, such as in [73, 104], or implicit, as in [54].

In models with explicit spatial interactions, the distance between agents influences agent behavior and interactions. For example, in the biological models mentioned above, space was explicit in the computations — agents reacted with other nearby agents and consumed nearby resources [55]. These distance-based interactions have shown up in agent-based approaches to location problems. In one study, agent-based models were used to locate bus stops in a continuous system [74]. Here, agents reacted to potential fields surrounding each location, thereby optimizing their own

position. The global fitness emerged from that ensemble of agent (bus stop) locations. Another study used a similar approach of attractive and repulsive forces between agents and demand centers to place retail establishments [73].

In all of the models mentioned in the previous paragraph [55, 73, 74], space was continuous. However, in the landscape problem presented in this thesis, the space is discrete. There are also examples in the literature of discrete agent-based approaches to location problems. In these examples, an agent's location is set using discrete grid coordinates. In one such study, a grid landscape was used to represent the spread of fire and evacuation of a building [97]. In this model, agents represented people; each cell contained a person or a building obstacle, or was unoccupied. At each time step, the fire spread according to observed patterns, and the agents made decisions about how to best escape the building. The model was used to evaluate how building obstacles and exits affected evacuation. In models with implicit spatial interactions, an agent's presence changes parameters on the landscape, which then leads to behaviors from other agents. An example of this is seen in [54], where the placement of agricultural crops influenced water runoff downstream. This change in runoff, in turn, influenced the placement of other crops selected to control runoff.

4.3 How models are used

ABMs are used for many purposes, including optimizing the layout of heterogeneous objects, evaluating the effects of different decisions of a diverse group of invested individuals, and reproducing observed behaviors in a population. The focus in this section is on examples that are similar to the ABM in this thesis.

4.3.1 Optimization

In agent-based models for optimizing layouts, the objective is the same as in any location optimization problem — to find the best place for a set of objects that maximizes the fitness of the system. In some cases, agent-based approaches have been applied to location problems that are typically solved with traditional methods (such as those covered in Chapter 3). Some exam-

ples include locating bus stops [74], demand centers [73], and facilities [4]. In other cases, the optimization problem included an additional constraint that fit well into an agent-based approach. For example, in one grid-based optimization problem, agents were used to optimize crop placements to minimize damage from water runoff [54]. Agents in this model represented different crop types with different runoff and profitability properties. A cellular automaton controlled water flow through the crop parcels; different crops had positive and negative effects on runoff. The model was tailored to a specific agricultural region and the types of crops grown in that area. It employed a Tabu-based search algorithm; crops in each cell exchanged locations to improve runoff on the landscape while still maintaining a desired profit. These swaps continued until the runoff score could not be improved.

This interaction between parcels over time cannot be easily captured in traditional methods, motivating the use of the agent-based approach. This same type of interaction is observed in the landscape problem in this thesis, where the presence of a plant in one cell affects the water resources in surrounding cells. One difference between the crop-placement problem and the landscaping problem is the presence of empty cells in the landscape problem. To generate a solution, the search algorithms applied to this problem swap plants in populated cells and move plants into empty cells.

4.3.2 Decision support

ABMs are also often used as a decision support tool, e.g. evaluating questions of natural resource use and management. In one such example, multi-agent simulations were used to study the viability of irrigated systems in a region in Africa [9]. Simulations included individual rules, collective rules, and the environment to evaluate the effects of different irrigation scenarios. ABMs can also inform land-use decisions, such as agriculture policy about agriculture in developing countries [11], the impact of land-use changes on wildlife populations [68], and the impact of human decisions on land-use and land-cover change [83]. Modeling development at the rural urban fringe was the focus of [16, 91]. In these studies, an ABM was used to explore the impact of greenbelts

on urban sprawl. Agents in the model were residents and service centers. Resident agents had different preferences for places to live; service centers needed to be located near population clusters. In this model, residents searched for a place to live by sampling a fixed number of cells and selecting those with the highest utility value, based on the resident's preferences.

4.4 Validation

Validation is the process of determining how well the model fits observed data [19]. Validating agent-based models can be particularly difficult for several reasons. These models are often developed for systems that are characterized by nonlinear interactions that occur over large spatial and temporal scales, making it difficult to collect enough data for empirical validation. These interactions can also be difficult to reproduce in the real world. Time and scale can also be issues in validating a model. For example, a model of the variables influencing global warming would be difficult to validate in a human lifetime. Models built to evaluate a dangerous situation can also be difficult to validate. In the fire evacuation model described in [97], it would be impossible to re-create the activities in the real world to study the effects of different floor plans on human survival rate.

Despite these difficulties, several quantitative and qualitative methods exist for validating agent-based models. One approach, called *docking*, aligns the agent-based model to another computational or mathematical model [5], showing that the two produce similar results. In [16] for example, a mathematical model and an ABM were developed to explore the effects of greenbelts on minimizing the ecological impacts of urban development. By comparing the results in simple cases and determining that both models produced similar outputs, the results of the ABM were effectively *docked*, establishing a starting point for more complex simulations.

Replicating empirical observations is another validation method for ABMs. In [6], an ABM reproduced annual fluctuations in agricultural production in the Long House Valley in Arizona from A.D. 200 to 1500 that were observed in paleoenvironmental data. An agent-based model of familial and individual nutritional requirements reproduced dynamics that followed food produc-

tion during this time period. As food became scarce, families moved around to follow the growing conditions, until eventually fewer and fewer families remained in the region. The declining population predicted by this model was also supported by the archeological record of Long House Valley population patterns.

Another approach to validation is to determine whether the model reproduces patterns known to be true heuristically, but for which empirical data may not exist. An ABM of bike racing strategies reproduced the physical dynamics observed in professional bike races [44], i.e. the ability of riders with varying skill levels to stay in a pack of riders. This model also reproduced the effects of different cooperation decisions among riders, where deciding to defect could have a negative effect on a rider's teammates. The claim was validated heuristically by an event that occurred in the 2009 Tour de France, where a rider's defection cost his teammate a podium finish.

This chapter introduced agent-based models and showed how ABMs are most often used for modeling systems where local interactions between items and their surroundings determine system performance. These interactions also exist in the landscape optimization problem, which is why an agent-based model is an appropriate choice for this problem. In all ABMs, agent behavior is defined by a set of rules. For the plant agents in the landscape model, the rules determine how the agents respond to light and water. How these rules were created for this work is described in the next chapter.

Chapter 5

Building the plant-growth model

Plants need light, water, and nutrients to grow. The plant-growth model that defines agent fitness in this study included two variables — light and water. Nutrients were not included as they can be controlled in built environments and are not part of the optimization criteria in this model. Data for this model was generated through experiments in a greenhouse using two plant species and different light and water conditions. These experiments, which were run in the Ramaley greenhouse at the University of Colorado, Boulder, were designed to answer the following questions:

- How do light levels affect plant growth?
- How do water application rates affect plant growth?
- How does plant spacing affect competition for water and the resultant plant growth?
- Do light levels affect water needs?
- How do light levels, watering rates, and spacing affect water application efficiency (WAE) for the plant species?

This chapter contains the details of these greenhouse experiments and their relationships to the plant growth model. The experimental setup used here was smaller than a real landscape, but still captured the relevant features of the model and the location problem. Plants had discrete locations, where each location received pre-determined, measureable amounts of light and water.

The effects of these conditions could be determined within a few weeks of growing, making it possible to run multiple trials of the experiments without waiting through entire growing seasons.

5.1 Experimental setup

Plants on residential landscapes often have heterogeneous light and water requirements. It would be impossible to include every landscape plant, or even every category of light and water requirements, in these experiments. Two plant species were used here — a full-sun, high-water species called *Tagetes patula* (marigold), and a shade-loving, high-water species called *Impatiens wallerana* (impatiens). Seeds were planted in 10 x 20 x 2 in. (25.4 x 50.8 x 5.8 cm.) trays covered with a mesh grid, as shown in Figure 5.1. Each cell in this grid was approximately 1 sq. in. (2.54 sq. cm.)¹ Two seeds were placed in each cell for redundancy to increase the likelihood that each cell had a plant. There were 9x24 cells in each tray, for a total of 216 cells and 432 seeds planted per tray of a single species. The trays were watered daily until the seedlings grew through the mesh, which occurred when they were approximately one inch high. Reaching this height took approximately three weeks for the marigolds and six weeks for the impatiens. The seedlings were then thinned so that each cell only had one seedling. The experiments described in the next sections were then begun.

The experiments were performed in series. The first one used the marigolds, which germinated and grew faster, to determine if this setup would produce the desired data. This experiment ran from September 24, 2011 to October 20, 2011; it was ended when the individual plants had outgrown their cells and were shading neighboring cells. After preliminary analysis showed that this marigold experiment was successful, the impatiens experiment was performed between January 12, 2012 to February 4, 2012 using the same approach. Each cell was planted with two seeds and watered daily until seedlings grew through the mesh. At that time, the seedlings were thinned and the experiments began.

¹ The mesh was plastic, and although advertised as having 1 sq. in. (2.54 sq. cm.) cells, the actual measurements were approximately 1 in x 3/4 in. (2.54 x 1.90 cm.)

At the conclusion of the growth periods, the biomass in each cell in each tray was recorded. To collect these values, individual plants were clipped at the base of their stems, dried, and weighed. The above-ground biomass is of the most interest in landscaping. In addition to total above-ground biomass in each cell, a separate measure of live biomass was recorded that included only those plants that were still alive at the end of the experiment. Root biomass was not collected due to the time and labor intensity of the process.

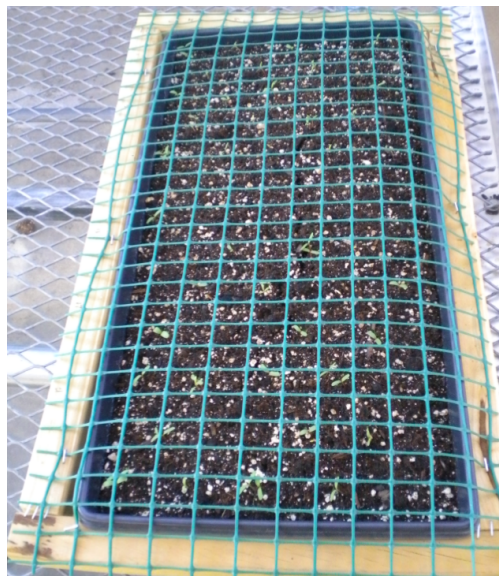


Figure 5.1: Tray with mesh grid used for planting. Each cell contained one plant, establishing evenly distributed, discrete locations for the plants in the experiment.

5.1.1 Determining how light levels affect plant growth

One of the key components of the plant-growth model in this thesis is how different plant species respond to light. To generate light-response curves for the plant agents, greenhouse experiments simulated varying light levels for individual cells in the trays by placing a screen in the middle of each tray, as shown in Figure 5.2. This screen changed the amount of light that some cells received throughout the day. The shade screen was constructed from burlap and a 6-in clear plastic shell. It generated a light reduction similar to that observed from real trees — approximately

70% — on real landscapes. Trays were aligned length-wise running east to west in the greenhouse. Each tray of marigolds also included two rows on the south side of the tray that were unaffected by the shading and received full sun all day. This approach provided directional shade protection — cells west of the shade screen were shaded in the morning, and cells east of the screen were shaded in the afternoon. Plants directly under the shade screen received shade all day. The cells receiving light (and shade) throughout the day were measured using the method described in the next section. The growth in each cell was then correlated to the light received by the plant in that cell throughout the experiment.



Figure 5.2: Experimental setup, including planting trays, mesh grid for controlling plant locations, and shade screen to generate variable light conditions.

The same approach was used for the impatiens. A different tray orientation was needed due to the lower sun angle in the month of January; all cells received full sun using the orientation for the marigolds, which made it impossible to determine how the plants grew in the shade. Cells on the south side of the impatiens tray received all-day sun, as shown in Figure 5.3. In this image, plants to the left of the shade screen (north) receive shade of varying degrees in the morning and afternoon, and plants to the right of the shade screen (south) receive sun all day. Just as with the marigolds, the growth in each cell at the end of the experiment was correlated to the light in that cell throughout the experiment.



Figure 5.3: Impatiens experiment showing the effects of shading on the landscape. Cells to the left of the shade screen were shaded and cells to the right of the shade screen received full sun all day.

5.1.2 Modifying light on the landscape

To determine how the shade screens affected the light in each cell on the landscape, the shaded cells were counted throughout the day. For the marigolds, counts were done on October 20, 2011 every 45 minutes between 10:30am and 4pm, which were the hours when the greenhouse received sun and was not shaded by surrounding buildings. If a cell was shaded at an observation time, the cell received a one and if not, the cell received a zero. Using this approach, the most heavily shaded cells received a value of 9, meaning they were shaded throughout the day, and the full sun cells received a value of zero, meaning they were never shaded. This approach generated the shade distribution shown in Figure 5.4, where the most heavily shaded cells are colored red and the full sun cells are colored blue. This process was repeated for the impatiens on February 4, 2012 between the same times. These shade values for both species were then partitioned into three categories of light availability. Areas shaded for less than one hour were labeled as full sun; areas shaded for between two and six hours were labeled as partial sun; and areas shaded for seven or more hours were labeled as full shade. Using this partitioning, only areas that receive sun for most of the day are full sun and only areas that receive shade for most of the day are full shade.

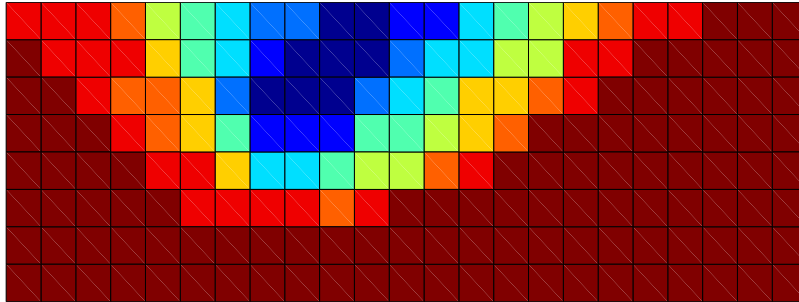


Figure 5.4: Map of the light distribution on the landscape in the marigold experiments. Available light was determined by counting the shaded cells at multiple times throughout the day. Blue cells are those that received full shade, and red cells are those that received all day sun. Cells with colors between red and blue received a mix of sun and shade.

5.1.3 Determining how watering frequency affects plant growth

Another component in the plant-growth model is how water resources affect growth. The water that a plant needs to grow is dictated by its environmental conditions and species-specific requirements. If this amount is not available, growth is reduced. This behavior is captured in the plant-growth model in the form of water-response curves, which were determined experimentally. The specific goals of these experiments were to:

- Determine the percentage reduction of growth under different amounts of water
- Correlate light conditions with water needs
- Determine how spacing affects competition for water, thereby impacting growth

Four watering frequencies were used — high, medium, low, and very low. High watering frequency trays were watered every day. Medium watering frequency trays were watered every other day. Low watering frequency trays were watered every three days, and very low watering frequency trays were watered every four days. This range ensured that the effects of water stress would be observed. At each watering, 1200mL of water was applied evenly to the tray. This amount saturated the soil profile in the trays watered every day, and water ran through the tray once the amount

applied exceeded the holding capacity of the soil. A second tray was placed beneath the planting tray to catch this excess water. Trays were weighed before and after watering every day, using the apparatus shown in Figure 5.5, to determine how much water had been lost through evapotranspiration since the previous day. The marigold experiment included two trays per watering level, for a total of eight trays. For the impatiens, there was only one tray per watering level due to limitations in the number of available seeds.²



Figure 5.5: Apparatus used to weigh trays. Trays were weighed each day before and after watering to determine water loss over the previous 24 hours.

5.1.4 Spacing and competition for water

In the light and water experiments just described, there was a plant in every cell in each tray. To determine how competition affected growth (biomass production) and water use, an additional experiment was performed with the marigolds where only every other row and column were planted. This arrangement, shown in Figure 5.6, provided more space for growth for individual plants. The same trays and watering schedules were used. This experiment started on January 23, 2012 and ran for 24 days, the same length of time as the original marigold experiment. Plants were

² Initially, coleus was selected as the shade species and there were enough seeds for a full experiment with eight trays. However, after multiple crop failures with the coleus, impatiens were used as a replacement.

started from seed, just as in that experiment. When the height of the plants reached the mesh grid, the experiment started. A shade screen was used to cover a portion of the tray; the other portion received full sun, as shown in Figure 5.7. This experiment was designed to support the component of the agent-based model that captures how access to surrounding water resources influences growth.



Figure 5.6: Marigolds planted in every other row and column in the competition experiment to determine how the additional space around each plant influenced its growth at different watering frequencies.

5.2 Data analysis and filtering

The growth response to the light and water conditions discussed above was determined from the mean biomass of the individual plants in each condition using:

$$mean_{bio} = \frac{\sum_{i=1}^n bio_i}{n} \quad (5.1)$$

where n is the number of populated cells, and bio_i is the biomass of plant i . In biological systems, outliers and plants growing on the edges of the system can exhibit behavior that skews the results. The biomass data collected here were filtered to minimize edge effects and remove outliers. To minimize edge effects, plants growing at the edges of the tray (first and last row and column) were omitted from the analysis, resulting in the following data set:

$$data' = data_{2:rows-1, 2:cols-1} \quad (5.2)$$

where *rows* is the number of tray rows, and *cols* is the number of tray columns. Outliers were defined on the basis of watering frequency. The mean and standard deviation were calculated for each watering frequency for each species. Plants with a biomass greater than two standard deviations from the mean were discarded:

$$data' = data < \mu(data) + 2\sigma(data) \quad (5.3)$$

The data set generated from these experiments included four values for growth response to water and three values for light response, for a total of 12 light/water combinations per species. These data were also used to determine the light and water interaction for each species, (i.e. how water use changes as a function of light, and the water application efficiency — biomass production in grams per mL of water applied.)



Figure 5.7: Setup of marigold competition experiment. Cells covered by the shade screen received full shade and the rest of the tray received full sun. Plants were spaced every other row and column.

5.3 Results

5.3.1 Germination

The marigolds had a high germination rate: between 88 and 96% per tray; as a result, most cells in each tray were populated. The percentage of populated cells for each tray is shown in Table 5.1. To conduct the experiments described in the previous sections, trays were randomly

assigned to different watering frequencies without considering plant sizes or cell coverage after germination. Unfortunately, the germination rates were significantly lower for the impatiens — between 52 and 73% — as shown in Table 5.2. Initially, there was difficulty getting any impatiens to germinate. After three weeks of daily watering in the Ramaley greenhouse with no germination, they were moved to the Macky greenhouse, which provided higher temperatures and humidity and constant moisture through a mist system. These conditions resulted in germination almost immediately, and the trays were then moved back to the Ramaley greenhouse. However, the overall germination rate may have been negatively affected by time spent in the Ramaley greenhouse. Most of the germination in these trays was along the edges with significant gaps in the cells in the middle of each tray.

Table 5.1: Cell coverage in all marigold trays. These numbers reflect the percentage of cells where at least one plant germinated.

Tray	Percentage of populated cells	Total plants
High water 1	86%	193
High water 2	96%	215
Medium water 1	88%	176
Medium water 2	96%	215
Low water 1	91%	204
Low water 2	92%	206
Very low water 1	93%	208
Very low water 2	94%	211

Table 5.2: Cell coverage in all impatiens trays

Tray	Germination Rate
High water	52%
Medium water	73%
Low water	63%
Very low water	57%

5.3.2 Growth response to water — marigolds

Figure 5.8 shows the mean live biomass vs. mean daily water applied for all four watering frequencies for the marigolds. As the watering frequency increases, so does the biomass. When

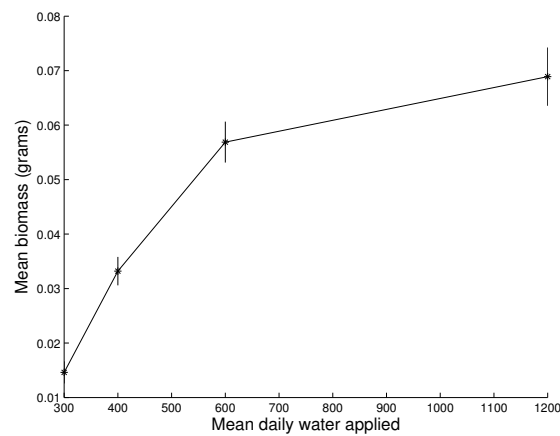


Figure 5.8: Mean biomass and the 95% confidence interval for marigold trays as a function of the water applied to the tray. Growth increased with increased water application up to the every other day watering frequency. Beyond this, the increase in growth leveled off for the amount of water applied as plants could not use all the water that was applied.

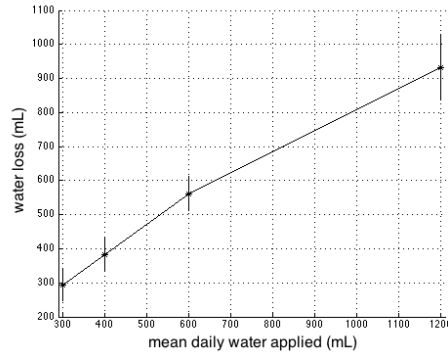


Figure 5.9: Mean water loss (with 95% confidence interval) as a function of the water applied at all four watering frequencies. The very low, low, and medium watering frequencies (300 mL, 400 mL, and 600 mL respectively) trays lost all of the water applied. The high watering frequency trays, however, lost only 933 mL of the 1200 mL daily.

watering frequency is decreased to every other day (600 mL), biomass production is also reduced to about 83% of the biomass in the daily watering. However, the water application of 600 mL also represents the maximum rate at which nearly all the water applied is also lost through evapotranspiration, as shown in Figure 5.9. This rate represents the most efficient use of water application. For the low watering frequency (400 mL), the biomass drops to 49% of the high watering frequency biomass, and for very low watering frequency (300 mL), the biomass is 21% of the biomass produced by daily watering. The details of each watering frequency are presented in the next sections, including how much water each tray lost and how the drying of the soil between waterings affected this water loss.

5.3.2.1 High watering frequency

Although 1200 mL of water was applied each day to the high watering frequency trays, those trays only lost a mean of 932.7 mL ($\sigma = 322.8$) a day through evapotranspiration. This water loss is calculated as the difference in the weight measured just after watering on one day and just before watering the next day, which includes both water use by plants through transpiration and water lost from the soil through evaporation. Water that ran through was collected and included in the daily

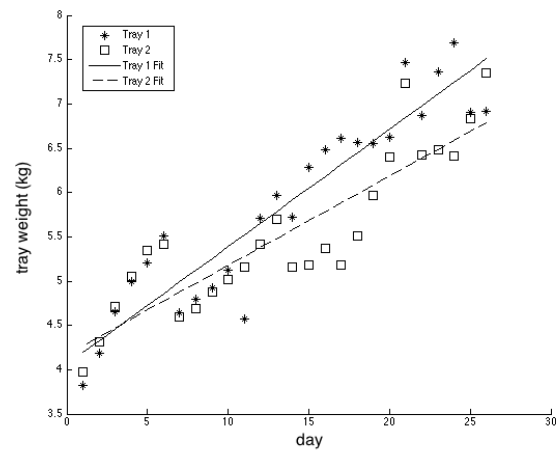


Figure 5.10: Tray weight over time for the high watering frequency marigolds. In these trays, the water applied was greater than the ET rate. As a result, the weight of the tray increased over time as the soil became saturated and water collected in the catch trays.

weight of the tray. The amount lost is well below the 1200 mL a day that was applied. As a result, the weight of both trays increased over time as the soil became saturated and water started collecting in the catch pans below the trays. The rate of increase can be described as:

$$weight = 0.1326 \cdot day + 4.0617, R^2 = 0.8681, \text{ for tray 1} \quad (5.4)$$

$$weight = 0.1007 \cdot day + 4.1696, R^2 = 0.7812, \text{ for tray 2} \quad (5.5)$$

where *weight* is the weight of the tray in kg and *day* is the day number of the experiment. This increase over time is also shown in Figure 5.10. The rate of increase is approximately 10-13% a day, which translates to 120-150 mL a day that is being applied to the tray, but not being used. This shows that the high watering frequency marigolds were not water-limited in their growth and establishes the maximum possible growth under ideal conditions in the plant growth model.

One component of the plant-growth model is the relationship between size and water. In this experiment, the size of the plants increased over time. Unfortunately, the water use in the high watering frequency trays did not increase with increasing plant size, as shown in Figure 5.11. Although there were temporary spikes in use, possibly due to changing environmental conditions in the greenhouse, no significant patterns emerged. There are several possible explanations for this result. An increase in transpiration by the plants could have been offset by a decrease in evaporation from the bare soil due to the increased leaf cover. It is also possible that evaporation was the dominant factor in this system and any increase in transpiration was too small, relative to evaporation, to affect the results. Therefore, although the model uses size as a factor in water use, examining the water-loss time series from these experiments did not produce the information needed to set that model parameter value. Other methods for correlating size and water use by modifying the spacing of the plants are discussed later in this chapter.

Validating water-use observations One common method for predicting water use is the Penman-Monteith (PM) equations detailed in Chapter 2. Those equations were used to validate the water-use behavior observed in these experiments. The greenhouse lets in approximately 80% of full sun, which is $1600 \frac{\mu\text{mol}}{\text{m}^2\text{s}^2}$ in the summer [61]. Assuming that the plants were approximately

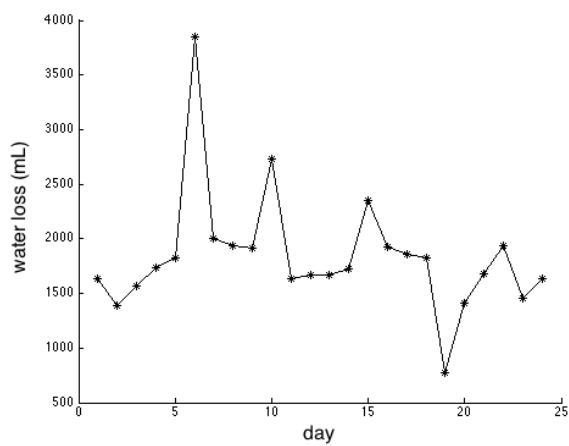


Figure 5.11: Total water loss each day in both high watering frequency trays. Each tray lost a mean of 932.7 mL daily: the mean for both trays was 1865.4 mL. There was no clear pattern in water loss over time to indicate that trays lost more water as plants increased in size.

four inches high with LAI values ranging between four and eight [69], a greenhouse temperature of around 25°C , and a 254 x 508mm tray, the PM equations yield a daily evapotranspiration amount of between 817 mL and 1009 mL. These experiments were performed in the fall, so we assume the sun was 75% of summer sun and get ET between 638 mL and 739 mL. These values are less than the 932.7 mL observed in the high watering frequency trays. However, both the LAI and the temperature are rough estimates. Increasing the estimated temperature by 2°C would increase water use to between 676 mL and 826 mL. In addition, these experiments were performed in two-inch-deep trays, which will have a higher evaporation rate than natural systems. Comparing the results here to the PM predictions shows that the observed water loss is comparable to expected values, providing support for the experimental results.

5.3.2.2 Medium water

For the medium watering frequency trays (watered every other day), the mean daily water loss was 562.1 mL a day ($\sigma = 171.1$), which is below the mean 600mL a day that was applied. A phenomenon observed here could explain why reducing water applied can also reduce growth. The mean water use the first day (the day the plants were watered) was 627.9 mL ($\sigma = 142.8$). However, on the second day, the water loss dropped to 521.4 mL ($\sigma = 159.1$), approximately 83% of the first-day water loss. A statistical significance test (two-tailed t-test) revealed that the difference in these means was statistically significant at the 95% confidence interval with a p-value of 0.02. As the soil dried out, the ET rate continued to drop. Without measuring each ET component individually, it is difficult to claim definitively whether the drop in ET was due to reduced evaporation, transpiration, or a combination. However, the reduced growth coupled with the reduced ET could indicate that, as the soil dried, plants may close their stomata to reduce transpirational losses.

The weight of the trays remained relatively constant over time, indicating that they were losing all of the water that was applied through ET. Just as in the high watering frequency trays, the water loss in the medium watering frequency trays did not increase over time. This showed

that water loss did not increase as the plants grew, which is relevant for modeling plant size and water loss described in the next chapter.

5.3.2.3 Low and very low water

In the low and very low watering frequency trays, water was applied every three or four days, respectively, providing ample time for the soil to dry out between waterings. The effects of reduced watering — reduced growth and reduced water loss with soil drying — were observed in these trays. The mean daily water lost in the low watering frequency trays was 383.8 mL ($\sigma = 165.1$), slightly below the mean daily 400 mL that was applied, as was shown in Figure 5.9.

5.3.3 Growth reduction from water stress — marigolds

The results show a strong correlation between the minimum water lost on the last day before watering and the reduction in total biomass. For example, the water lost in the medium watering frequency trays on the second day was 83% of water lost on the first day. Looking back at Figure 5.8, one can see that the growth in the medium watering frequency trays was about 83% of growth in the high watering frequency trays. This pattern continued for the other watering frequencies as well. For example, water loss declined in the low watering frequency trays to 45% of first day values on the final day before watering, and biomass in the low watering frequency trays was only 45% of what it was in the high watering frequency trays.

Figure 5.12 graphs the observations discussed in the previous paragraph. The x-axis shows the percentage of the maximum mean daily water loss for a watering frequency on the last day before watering. For high watering frequency trays that are watered every day, this number is 1.0. For very low watering frequency trays that dry out for three days, this value is 0.45. On the y-axis is the percent of the maximum mean biomass in all trays. The maximum mean biomass is found in high watering frequency trays — 0.069 grams. The mean biomass values in the medium, low, and very low watering trays are 0.057 grams, 0.033 grams, and 0.015 grams respectively. These numbers are 83%, 49%, and 22% of 0.069. These percentages are also very close to the reductions

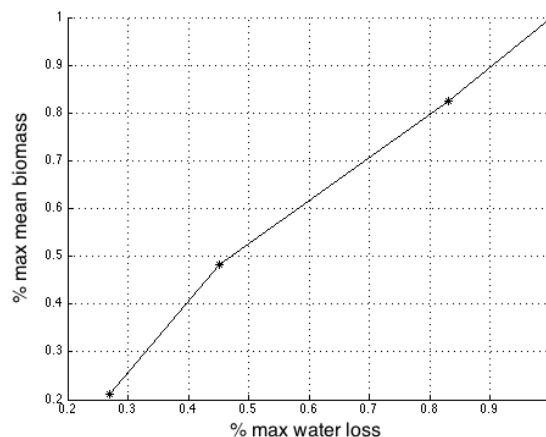


Figure 5.12: The percentage of the maximum mean biomass produced in all marigold experimental conditions as a function of the minimum percentage of water consumed in a tray.

in water loss at these watering frequencies.

The details about water reductions and soil moisture provided above are meant to illustrate how applying less water can lead to reduced growth. In the plant growth model, the end result of reduced watering is captured, but not necessarily the role of soil moisture in determining the growth reduction.

5.3.4 Growth response to water — impatiens

Figure 5.13 shows the mean live biomass vs. mean daily water applied in all four watering frequencies for the impatiens. Unlike the marigolds, these plants did not perform well when they were watered every day. Rather, growth increased steadily for the very low, low, and medium watering frequencies, then dropped off for the high watering frequency to 76% of the growth in the medium watering frequency. There was a clear visual difference between the two trays on the final day of the experiment, as shown in Figures 5.14 and 5.15. While it may seem surprising that growth could actually drop with increased water application, a literature search revealed that impatiens are sensitive to over-watering [7].

Interestingly, for the low watering frequency, the biomass was also 76% of the medium

watering frequency biomass. For the very low watering frequency trays, biomass dropped to 60% of the amount observed in the medium watering frequency trays. Just as with the marigolds, an important detail in these experiments was the relationship between ET, soil moisture, and growth. The details for each watering frequency and the effect on growth are discussed individually below.

5.3.4.1 High water

The mean daily water loss in the high watering frequency trays was 883.1 mL ($\sigma = 455.5$), close to the 932.7 mL observed in the marigold high watering frequency trays. However, the mean biomass for the impatiens was about 1/3 the biomass of the high watering frequency marigolds. Just as with the marigolds, this water loss was well below the 1200 mL applied daily, which resulted in an increasing tray weight throughout the experiment as the soil became saturated and water collected in the catch pan. The increase in weight can be described as:

$$weight = 0.1267 \cdot day + 7.7807, R^2 = 0.7830 \quad (5.6)$$

There was also no correlation between the daily water loss and plant size. Water loss on each day is shown in Figure 5.16. There were temporary spikes in loss, just as with the marigolds, possibly due to changing environmental conditions in the greenhouse.

5.3.4.2 Medium water

The mean daily water loss in the medium watering frequency trays was 592.2 mL ($\sigma = 259$). Unlike the high watering frequency tray, where the weight of the tray increased over time as water was collecting in the catch pans due to overwatering, the weight of the medium tray remained constant, showing that all the water being applied was being lost through evapotranspiration. Another similarity to the marigolds was the pattern of higher ET on the day water was applied. The mean daily water loss on the day of, and the day after, watering were 691.7mL ($\sigma = 262.8$) and 483.6mL

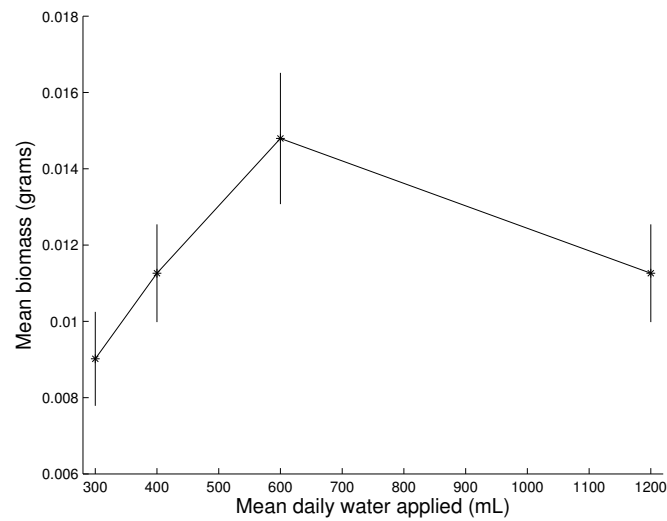


Figure 5.13: Mean biomass with the 95% confidence interval for the impatiens as a function of the amount of water applied. The x-axis in this graph is the mean daily water applied, where 1200 mL was the high watering frequency, 600 mL was the medium watering frequency, 400 mL was the low watering frequency, and 300 mL was the very low watering frequency. Unlike the marigolds, where growth increased with additional water applied, impatiens grew less in daily watering than when they were watered every other day.



Figure 5.14: Impatiens watered every day (high watering frequency) on the final day of the experiment. These plants did not perform as well as the impatiens watered every other day.



Figure 5.15: Impatiens watered every other day on final day of the experiment. This species performed better with watering every other day than with daily watering. This image also shows the effects of shading. Plants to the left of the shade screen that received shade or partial shade performed better than plants to the right of the screen, which received full sun throughout the day.

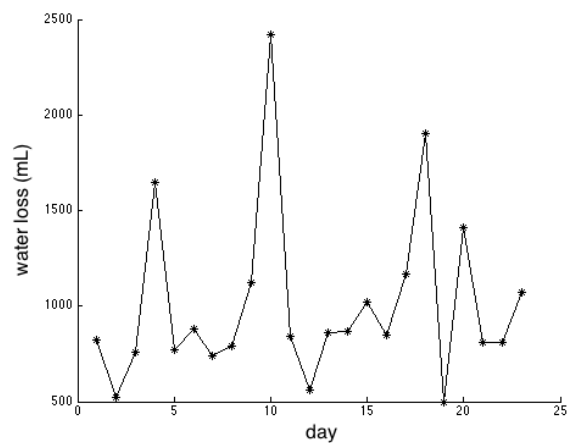


Figure 5.16: Total water lost each day for the impatiens in the high watering frequency tray. Just as with the marigolds, there was not a clear increase in water loss over time as plants increased in size.

($\sigma = 216.2$) respectively. Water loss on the day after watering was approximately 70% of the water loss on the day of watering.

5.3.4.3 Low and very low water

The impatiens in the low and very low watering frequency trays showed the same pattern as the marigolds in these frequencies — reducing the water applied also reduced biomass production, with a drop in water loss each day that plants were not watered. The mean daily water lost in the low watering frequency trays was 430.0 mL ($\sigma = 205.2$), slightly above the mean daily 400 mL that was applied. While it does not seem possible that trays could lose more water than what was applied, the pattern observed here can be explained by the decreasing tray weight over time, as shown in Figure 5.17. The existing water resources in the tray were being lost to supplement the water being applied. Just as in the medium watering frequency trays, there was a pattern of decreasing water loss with decreasing soil moisture. On the day of watering, the mean water loss was 545 mL ($\sigma = 127.8$). The day after watering, this mean dropped to 500 mL ($\sigma = 179.4$); it dropped again to 218.6 mL ($\sigma = 146.5$) two days after watering.

In the very low watering frequency trays, the mean daily water loss was 360 mL ($\sigma = 247.7$), slightly above the 300 mL daily that was applied. Just as with the low watering frequency trays, losing more water than was applied can be explained by the weight of the tray decreasing throughout the experiment. Plants were using existing water in the soil in addition to the water being applied. As this existing water was used up, the weight of the tray decreased to indicate that the soil was drying out. The weight drops for the first 12 days, then stabilizes. The mean water loss after the tray weight stopped declining was 277.5 mL ($\sigma = 201.6$), just below the 300mL a day that was applied.

The drop in water loss with each day that plants were not watered was also observed in the very low watering frequency trays. Water loss on the day of watering was 646.7 mL ($\sigma = 173.7$) a day. The day after watering, daily water loss was 293.3 mL ($\sigma = 216.1$). On the third and fourth days after watering, mean daily water loss was 240 mL ($\sigma = 133.6$) and 240 mL ($\sigma = 222.6$)

respectively.

5.3.5 Growth response to light — marigolds

The marigolds' growth response to light shows how the interaction between light and water affects growth. The mean growth at all three shade levels (full sun, partial sun, and shade) for all watering frequencies can be described by:

$$growth = -0.0063 \cdot shade + 0.0521, R^2 = 0.99 \quad (5.7)$$

where *growth* is biomass in grams, and *shade* $\in [1, 2, 3]$. Growth decreases with increasing shade, which is not surprising — marigolds are full-sun plants that need light to grow and reducing the light can reduce growth. The variability in growth within each light level decreases as the shade increases, suggesting that plants in the low-light conditions were light stressed and increasing the water would not increase growth. The largest confidence interval on the three means occurs in the data for the full-sun conditions — the confidence intervals and p-values are shown in Table 5.3. All means are statistically significant at the 95% confidence interval. The means shown here include the growth at all four watering frequencies, suggesting that the watering frequency had a strong effect on growth in full sun and generated more variability in the results.

To illustrate the interaction of shading and water, the growth in the individual watering frequencies as a function of light is shown in Figures 5.18. In the trays that received daily water, the highest growth observed was in full sun, and the growth dropped with increasing shade:

$$growth_{hw} = -0.0162 \cdot shade + 0.0914 \quad (R^2 = 0.99) \quad (5.8)$$

This pattern is also observed in the medium watering frequency trays. However, the slope of the decline as shade increases is not as severe as in the high watering frequency trays:

$$growth_{mw} = -0.0149 \cdot shade + 0.0769 \quad (R^2 = 0.99) \quad (5.9)$$

In the low watering frequency trays, reducing light had a slight effect on growth.

$$growth_{lw} = -0.0016 \cdot shade + 0.0353 \quad (R^2 = 0.96) \quad (5.10)$$

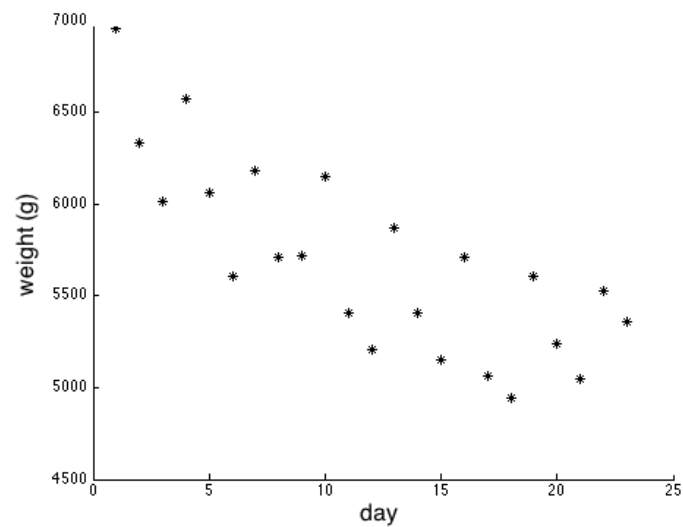


Figure 5.17: Tray weight (grams) over time for low watering frequency tray. The weight of the tray decreased throughout the experiment, indicating that the soil was drying out. This decreasing weight also explains why ET was higher than the amount of water applied for the first 12 days of the experiment. There were existing water resources in the tray that were being depleted, and the weight of the tray decreased as the soil dried out.

Table 5.3: Confidence intervals and p-values for mean growth in the three shade levels.

Shade level	CI range (g)	p-value
1	0.0002 to 0.0908	0.0494
2	0.0125 to 0.0673	0.0189
3	0.0223 to 0.0434	0.0022

In the very low watering frequency trays, the pattern reverses, and plants produced more biomass growing in the shade than in the sun:

$$growth_{vlw} = 0.0073 \cdot shade + 0.0048 \quad (R^2 = 0.96) \quad (5.11)$$

In these very low water conditions, the growth reduction from drought stress overwhelmed the effects from reduced light. The plants in the shade had protection that reduced their water requirements and enabled them to survive, while those in full sun conditions were not so protected. This was the only evidence in these experiments of facilitation, where shading enabled a plant to survive in drought conditions. The plant-growth model in this thesis also includes facilitation, since it is such an important element in modeling plant interactions on a landscape.

Another way to look at the interaction between light, water, and growth is to examine biomass production at each watering frequency in different light levels. Figure 5.19 shows how reduced watering frequency caused a greater growth reduction as light increased. Plants in full sun conditions experienced a considerable growth reduction under reduced watering. In fact, there is an 86% drop in growth in these conditions between the high watering frequency trays and the very low watering frequency trays. As the shade increases, the growth at the different watering frequencies is much less variable, as indicated by the narrower confidence interval. In these conditions, the drop in growth between the highest and lowest watering frequency is approximately 37%. This relationship can be described as:

$$growth_{l1} = 0.0001 \cdot water + 0.0073 \quad (R^2 = 0.75) \quad (5.12)$$

where $water \in [300, 400, 600, 1200]$ and $growth_{l1}$ is biomass produced in full-sun conditions, and

$$growth_{l2} = 0.00004 \cdot water + 0.0145 \quad (R^2 = 0.91) \quad (5.13)$$

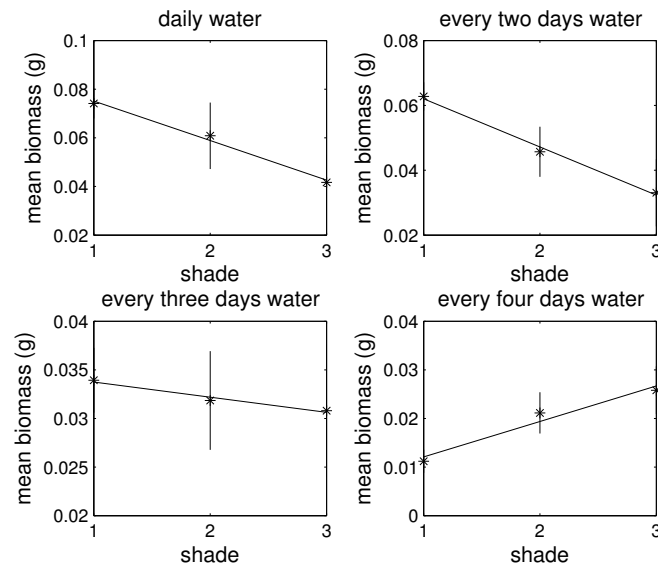


Figure 5.18: Mean biomass with the 95% confidence interval for the marigolds in the three shade conditions (shade = 1,2,3 for full sun, partial sun, and shade, respectively) for each individual watering frequency. This graph shows that the reduction in growth due to light reduction depended upon how much water the plants received. When the plants were watered every day or every other day, they experienced more of a growth reduction in low light (shade = 3) than when they were watered less frequently. In the very water-limited situation (every four days), reducing light increased growth as plants in the shade benefitted from the reduced water requirements for survival in these conditions.

where $growth_{l2}$ is biomass produced in partial-sun conditions, and

$$growth_{l3} = 0.00002 \cdot water + 0.0228 \quad (R^2 = 0.95) \quad (5.14)$$

where $growth_{l3}$ is biomass produced in shade conditions.

The light, water, growth relationship can also be described using multilinear regression:

$$growth = \beta_0 + \beta_1 \cdot shade + \beta_2 \cdot water + \beta_3 \cdot shade \cdot water \quad (5.15)$$

where $growth$ is biomass in grams, $shade \in [1, 2, 3]$ and $water \in [300, 400, 600, 1200]$. Solving for $\beta_0, \beta_1, \beta_2$, and β_3 yields:

$$growth = 0.0125 + 0.0011 \cdot shade + 0.00006 \cdot water - 0.000004 \cdot shade \cdot water \quad (5.16)$$

with an $R^2 = 0.64$.

The reduction in biomass at different shade and water levels is incorporated into the plant growth model to balance trade-offs between light and water for optimizing water application efficiency.

5.3.6 Growth response to light — impatiens

The impatiens were selected for these experiments because they are a shade species, which means that they require protection from full sun conditions in order to survive, and should have growth curves that look quite different from the marigold growth curves in the same light [7]. While the marigolds grew best in full sun in these experiments, the impatiens struggled in full sun and grew best in full shade. Using the shade categories ($shade = 1, 2, 3$ for full sun, partial sun, and shade respectively), the growth at each watering frequency for each of these light levels was calculated. The results of this calculation are shown in Figure 5.20. For all watering frequencies except the low one, growth decreases with increasing light, which is opposite of the pattern observed for the marigolds. The trend is strongest for the high and medium watering frequencies.

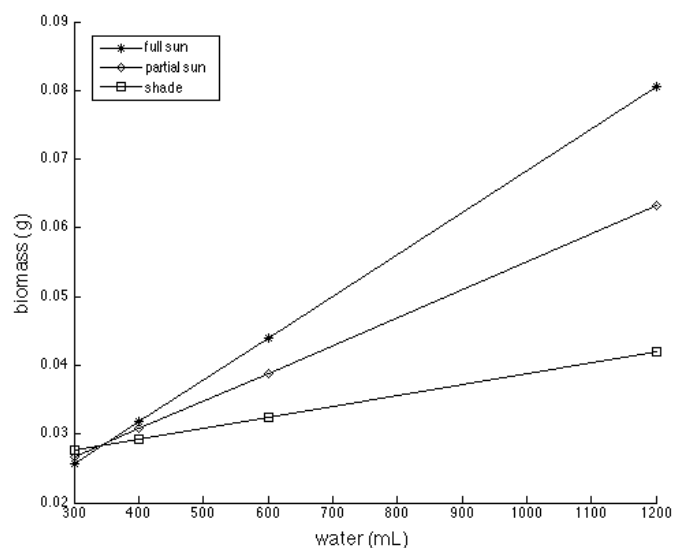


Figure 5.19: Mean growth vs. watering frequency in full sun, partial sun, and shade conditions. The x-axis here is the mean daily water that was applied, where 1200 mL was the high watering frequency, 600 mL was the medium watering frequency, 400 mL was the low watering frequency, and 300 mL was the very low watering frequency. In shade conditions, reduced water had less of an effect on growth than it did in full sun conditions.

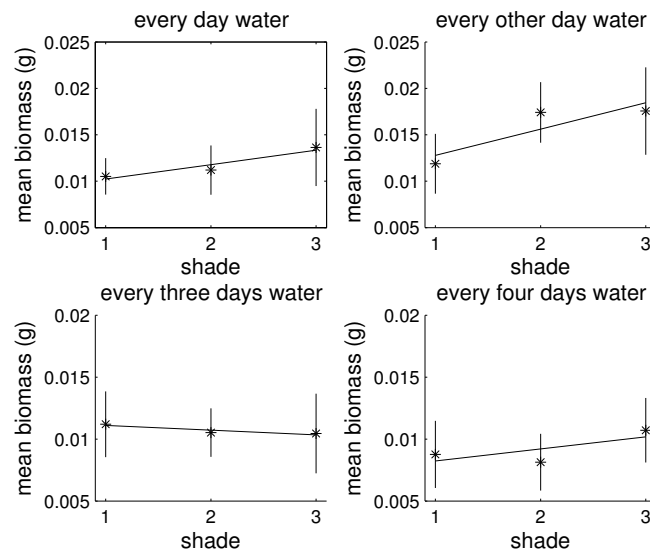


Figure 5.20: Mean growth with 95% confidence interval vs. light for all four watering frequencies for the *impatiens*. The shade categories here are full sun, partial sun, and shade (shade = 1, 2, 3, respectively). Unlike the marigolds, these plants grew better in the shade than in the full sun. The effect of the light was the most dramatic in the medium watering frequency (every other day), which was the optimal frequency for this species.

The relationship between growth and light for each watering frequency can be described as follows:

$$\begin{aligned}
 growth_{ihw} &= 0.0016 \cdot shade + 0.009 \quad (R^2 = 0.90) \\
 growth_{imw} &= 0.0028 \cdot shade + 0.010 \quad (R^2 = 0.77) \\
 growth_{ilw} &= -0.0004 \cdot shade + 0.011 \quad (R^2 = 0.83) \\
 growth_{ivlw} &= 0.001 \cdot shade + 0.007 \quad (R^2 = 0.52)
 \end{aligned} \tag{5.17}$$

where $growth_{ihw}$ is biomass produced in the high watering frequency, $growth_{imw}$ is biomass in the medium watering frequency, $growth_{ilw}$ is biomass in the low watering frequency, $growth_{ivlw}$ is biomass in the very low watering frequency, and $shade \in [1, 2, 3]$ represents full sun, partial sun, and shade conditions respectively.

5.3.7 Spacing and competition for water

One of the critical components of the agent-based model is the relationship between the plant agent's size and the agent's ability to access water resources beyond its home cell on the landscape — e.g. larger plants can access more water due to a more extensive root structure. In the greenhouse experiments, all plants were grown from seed and the sizes were relatively uniform at the beginning of the experiment. At the end of these experiments, there were differences in size within each tray, as previously described. In addition, each tray had its share of outliers: plants whose biomass was more than two standard deviations above the mean for the tray.

One possible explanation for the size variability is that large plants were able to out-compete their neighbors for water. A photo of the root growth for half of the tray of the high watering frequency marigolds is shown in Figure 5.21. It is clear that plants grew beyond their individual cells. An exact correlation of plant size with the extent of root structure is beyond the scope of this thesis. However, spatial auto-correlation was applied to each cell to determine if larger plants were

thriving at the expense of their neighbors. The following auto-correlation formula was used:

$$\begin{aligned}
 B &= (b_{xy}), x = 1 \dots n, y = 1 \dots m, \text{ biomass of plant at } x, y \\
 n &= \text{landscape rows,} \\
 m &= \text{landscape columns} \\
 BA &= (ba_{xy}) = \mu(b_{x-1:x+1, y-1:y+1}) - b_{xy}, x = 1 \dots n, y = 1 \dots m \\
 \rho &= \frac{E[B - \mu_B](BA - \mu_{BA})}{\sigma_B \sigma_{BA}}
 \end{aligned} \tag{5.18}$$

where b_{xy} is the biomass of the plant at (x, y) , ba_{xy} is the mean biomass of plants in a one-cell radius of (x, y) , and ρ is Pearson's correlation coefficient between b_{xy} and ba_{xy} . The Pearson correlation returns a value between -1.0 and 1.0, where a -1.0 indicates a negative correlation and 1.0 indicates a positive correlation. Here, a negative score indicates that a large plant is surrounded by smaller plants, which could be due to competition. The larger plants were extracting water from neighboring cells, causing a growth reduction for these neighboring plants. A positive score indicates the opposite, i.e. plants were benefiting from their neighbors and sizes were consistent within a cluster.

The ρ value was calculated for each tray individually. The calculations included the outliers, but not the edge plants. The outliers were included because these were the largest plants in the tray and the objective was to determine if large plants were affecting their neighbors. Edge plants were not included because they had fewer neighbors, which could skew the results. There was not a significant correlation between the biomass of individual plants and the mean biomass of their neighbors, with one exception. In the very low watering frequency trays, there were positive correlations of $\rho = 0.40$ and $\rho = 0.65$ in tray 1 and tray 2, with p-values of $8.0542e-07$ and $1.4386e-18$, respectively. This positive correlation is not a sign of competitive advantage for larger plants, but rather a sign of consistent growth in different light conditions. Plants in good growing conditions grew well and were surrounded by other plants that also grew well (relative to the growth in the tray), while plants in poor conditions suffered, as did their neighbors.

To capture competitive effects that cannot be observed by planting every cell in a tray, an additional experiment was performed with the marigolds. The specific goal was to explore the



Figure 5.21: Roots growing along the bottom of one of the trays for the high watering frequency marigolds. This image shows approximately half the tray. Clearly, these plants did not restrict their growth to the cells in which they were planted and surrounding cells

effects of reducing the planting density on competition and water use. Plants were spaced every other row and column so that all plants were surrounded by empty cells. The same four watering levels were used, with 1200 mL applied at each watering. An eight-inch shade screen was used to cover half the tray to generate two levels of shading — full sun and full shade.

There were far fewer plants in the high watering frequency than in the others due to a problem with one of the high watering frequency trays. A tray without holes was used and water collected in the tray instead of draining into the catch tray, until the tray flooded. The data from that tray is not used here.

The effect of the additional spacing on growth depended upon the watering frequency and light conditions for the plants. In the shade, the additional space did not produce a statistically significant difference in growth. In both situations, plants were light stressed and the additional water was not needed. The same was true for the full sun, high watering frequency conditions: the additional space did not increase growth. In this case, plants were receiving enough water: the additional water that could be obtained due to reduced competition did not increase growth.

The benefit of additional spacing became apparent when the watering frequency was reduced. Growth in full sun conditions in the medium, low, and very low watering frequency trays increased significantly when plants had additional spacing. Figure 5.22 shows a clear separation in growth that started with the medium watering frequency and widened for the low and very low watering frequencies. For the medium and low watering frequencies, a 50% increase in growth was observed with increased spacing. For the very low watering frequency trays, the increase was almost 500%. The importance of this result to the agent-based model is that decreasing competition for water allows plants to survive in lower watering frequency conditions. This result is incorporated into the model via competition for water around each plant agent.

5.3.8 Water application efficiency

The water application efficiency (WAE) is defined here as the biomass produced in grams per mL of water applied. The WAE calculation included all plants in a tray including the edge

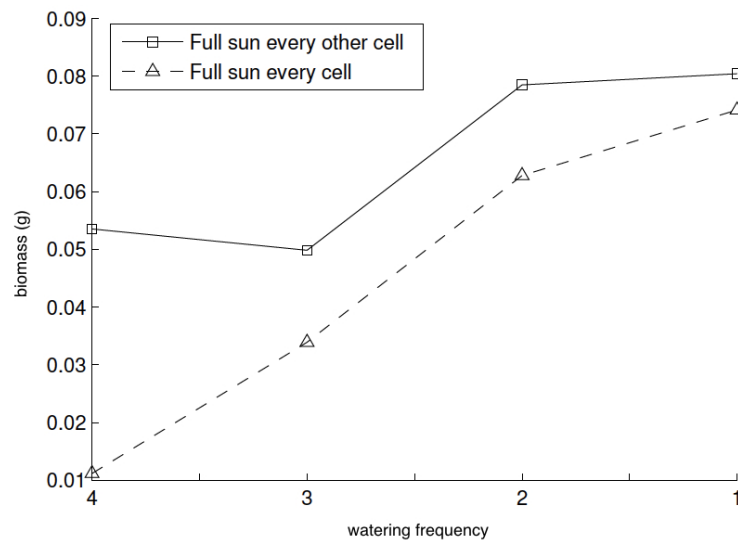


Figure 5.22: Mean growth in full sun when marigolds were planted in every cell in a tray and every other cell in a tray. In high watering frequency (watering frequency = 1), full sun conditions: increasing the space around each plant did not significantly increase growth. However, in full sun with medium, low, and very low watering frequencies (watering frequency = 2,3,4), the additional space did increase growth. This was particularly true in the very low watering frequency.

plants and the outliers that were filtered out for other calculations. (Since edge plants and outliers use water, not including them would skew the calculations by varying degrees, depending on the watering frequency.) The formula for calculating WAE for a watering frequency is:

$$WAE = (\mu_{gr} \cdot pc) / water \quad (5.19)$$

where μ_{gr} is the mean growth in the trays for the watering frequency, pc is the number of populated cells for the watering level, and $water$ is the daily water applied to the trays. For example, to calculate WAE in the high water marigold trays, the equation would be:

$$WAE_h = 0.076 \cdot 408 / 2400 \quad (5.20)$$

5.3.8.1 Marigolds

The WAE values obtained for the marigolds are shown in Figure 5.23, where the x-axis is the mean daily water applied and the y-axis is g/mL for that watering frequency. Peak WAE is found in the medium watering frequency trays, which received 600 mL daily. There are two data points in this graph for the high watering frequency trays — the mean water lost in these trays (932.7 mL) and the water applied (1200 mL). The water lost was included here due to the overwatering in these trays: plants could have survived on less water than what was being applied, and therefore, the water applied may not be the best metric for WAE. When WAE was calculated using water lost, WAE for the high watering frequency trays was lower than the WAE for the medium trays.

WAE in different light levels within the same watering frequency is also important. Although the water applied to each tray was constant, the biomass produced under different light levels within each tray changed. The WAE at the three shade levels within each watering frequency is shown in Figure 5.24. The pattern in this figure is, as expected, similar to the relationship between biomass and light at each watering frequency. Since WAE is directly tied to biomass production, if plants grew less in shaded conditions, but the same amount of water is applied, their WAE will be lower than in full sun. For high watering, this relationship is:

$$WAE_h = -0.0004 \cdot shade + 0.0018 \quad (R^2 = 0.71) \quad (5.21)$$

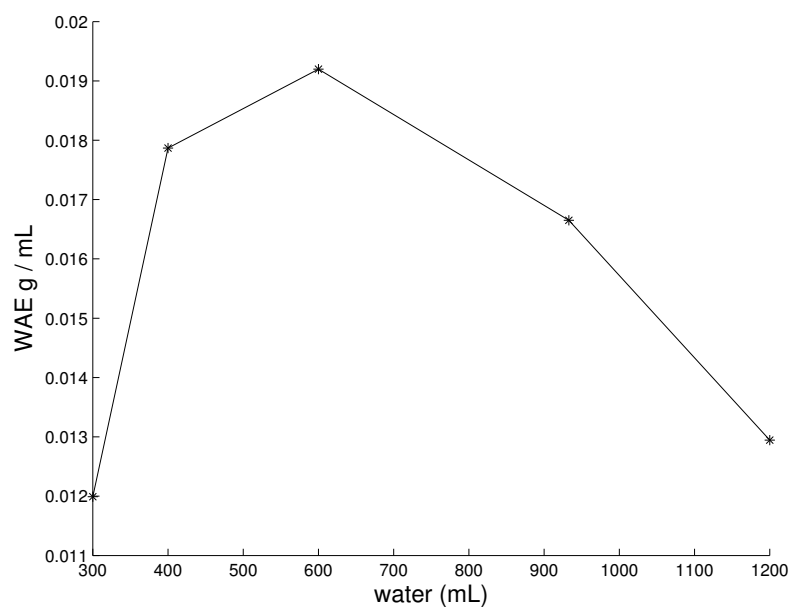


Figure 5.23: WAE in g/mL for the marigolds in all watering levels. The x-axis is the daily water applied to a tray for a watering frequency and the y-axis is the grams/mL. Although the marigolds had the highest biomass in the high watering frequency, WAE peaked at the medium watering frequency. This was the frequency that produced the most biomass for the water applied.

For high water, based on consumption:

$$WAE_{hc} = -0.0005 \cdot shade + 0.0023 \quad (R^2 = 0.71) \quad (5.22)$$

For medium water:

$$WAE_m = -0.0006 \cdot shade + 0.0028 \quad (R^2 = 0.89) \quad (5.23)$$

For low water:

$$WAE_l = -0.0008 \cdot shade + 0.0034 \quad (R^2 = 0.95) \quad (5.24)$$

For very low water:

$$WAE_{vl} = 0.0003 \cdot shade + 0.0021 \quad (R^2 = 0.61) \quad (5.25)$$

5.3.8.2 Impatiens

For the impatiens, there was a clear drop in WAE between the medium and high watering frequency trays due to a combination of decreased growth, fewer populated cells, and twice as much water applied as in the medium watering frequency. The WAE for all four watering frequencies is shown in Figure 5.25. The WAE for the medium, low, and very low watering frequencies are clustered together, while the WAE for the high watering frequency was significantly lower. It is interesting to note that the impatiens in the low and very low watering frequencies did not experience the same mortality as the marigolds at these watering levels. This result could be due to species differences or changing conditions in the greenhouse — it was colder in February than it was in October.

The experiments described in this chapter were performed to build and test the model described in the next chapter. The features of the model include how two plant species, represented as agents with different growth requirements on a simulated landscape, respond to simulated light and water resources, and how competition for water affects growth.

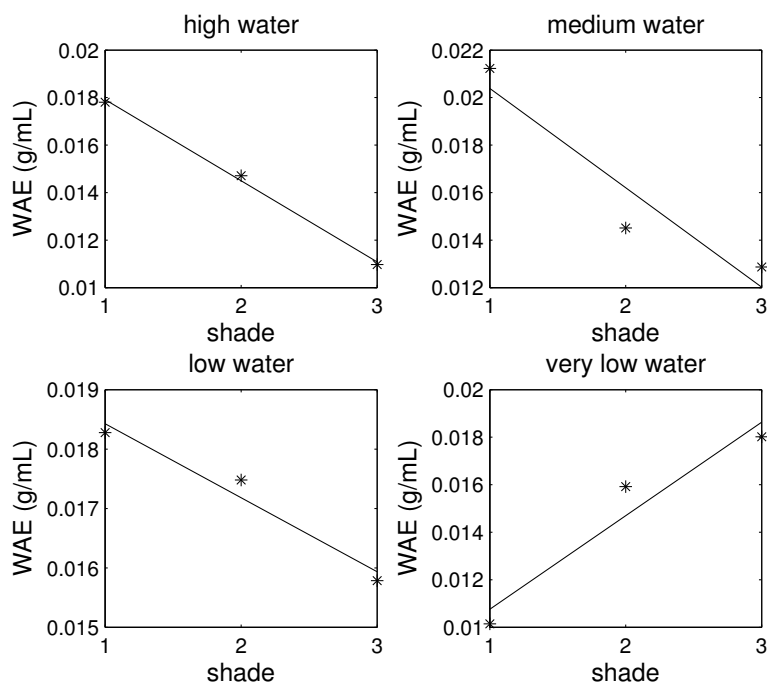


Figure 5.24: The WAE in g/mL for the marigolds varied for the three light categories for each watering frequency. In the high, medium and low watering frequencies, WAE was lower in the full shade (shade = 3) than in the full sun (light = 1). For the very low watering frequency, WAE increased in the full shade (shade = 3), possibly due to a 67% mortality rate in full sun as compared to a 4% mortality rate in full shade.

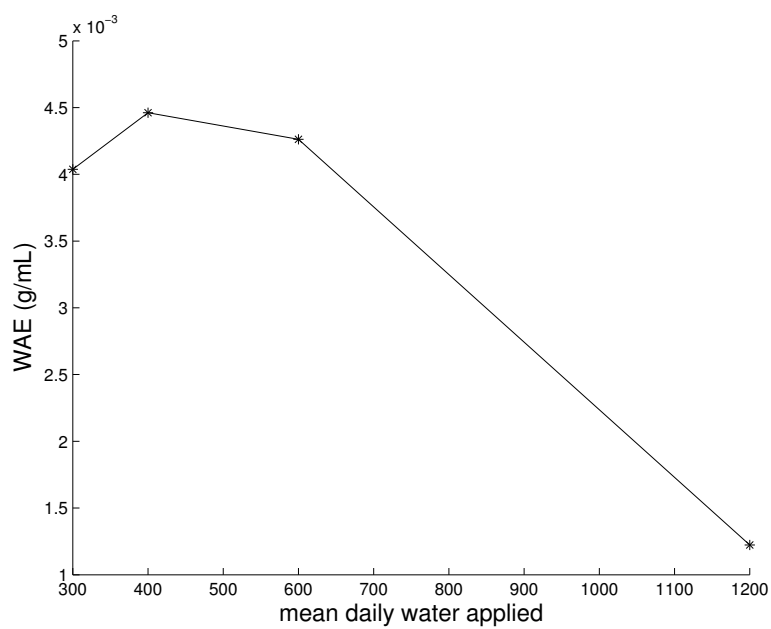


Figure 5.25: WAE in g/mL forimpatiens for all watering frequencies. The highest observed WAE was in the low watering frequency for this species. However, the WAE for the medium, low, and very low watering frequencies were all similar.

Chapter 6

An agent-based model

On residential landscapes, the primary factors that influence plant growth are light, water, and nutrients. Light and water are likely to be the most variable due to light interception from trees and to uneven irrigation and competition for water. Nutrients can also vary, but these can also be easily modified and they are not part of the optimization criteria. The agent-based model described in this chapter is designed both to model growth of individual plants in varying light and water conditions and to discover water-conserving arrangements that enable plants to flourish while still using water efficiently. In this model, each plant is an agent that responds to the available light and water resources on the landscape. The specific details of the response are captured in a plant-growth model that uses data collected in the greenhouse experiments explained in the previous chapter. A multiobjective fitness function captures how well the available light and water resources match the agent's light and water requirements as well as how efficiently the agent uses the water. This fitness function plays a key role in the optimization strategy that is described in the next chapter.

The significant components of the agent-based model include:

- **Landscape** — Represents light and water conditions, plant-agent locations, and the results of plant-agent interactions with each other and their surroundings.
- **Plant agents** — Represent individual plants that “live” on the landscape and respond to the light and water conditions.

- **Plant growth model** — Establishes how plant agents grow under different light and water conditions.
- **Fitness function** — Measures how well the light and water resources at a location match a plant agent's light and water requirements. This function also captures water application efficiency (WAE) at a particular location.

The landscape and plant agent properties are shown in Figure 6.1. In this chapter, each of these properties is described and related back to specific results from the greenhouse experiments.

6.1 Landscape

The landscape is represented as a discrete grid, similar to the one used to set plant locations in the greenhouse experiments:

$$\begin{aligned}
 L &= (l_{xy}), x = 1 \dots n, y = 1 \dots m, \\
 n &= \text{landscape rows} \\
 m &= \text{landscape columns}
 \end{aligned}
 \tag{6.1}$$

Each cell has two parameters:

$$\begin{aligned}
 LI &= (li_{xy}), x = 1 \dots n, y = 1 \dots m \\
 WATER &= (water_{xy}), x = 1 \dots n, y = 1 \dots m
 \end{aligned}
 \tag{6.2}$$

where LI is the light and $WATER$ is the water. These values represent the amount of the corresponding resource that is present in the cell. In this model, light is defined the same way as it was in the greenhouse experiments, as the level of light reduction, or shading, and takes on three discrete values — full sun, partial sun, and full shade, where $LI \in [1, 2, 3]$ respectively. These values are for LI are the same numbers used in the greenhouse experiments to represent how often a cell is shaded throughout the day. The water value includes the frequency and amount of water applied:

$$water = \frac{\alpha}{wf}
 \tag{6.3}$$

where α is the amount applied at each watering and wf is the number of days between waterings. For the model presented in this thesis, $\alpha = 1200 \text{ mL}$, and $wf \in [1...4]$, which are the same values used in the greenhouse experiments.

The size of each cell is a property of the landscape (cell size) and is adjustable. Greenhouse experiments used a cell size of approximately one square inch. A plant agent's location on the landscape is a discrete l_{xy} , and only one agent can occupy a cell at a time.

A phenomenon not reproduced in the greenhouse experiments described in this thesis, but observed on real landscapes, is the difference between morning and afternoon light [111]. For instance, an ideal morning location might present harsh afternoon conditions. This feature is not included in this thesis, but could be implemented by dividing light into two categories — morning and afternoon — and updating the landscape properties to have two light parameters: ML (morning light), and AL (afternoon light).

6.2 Plant agents

In the plant-agent properties shown in Figure 6.1, each agent has a light and water requirement category that determines how it responds to the corresponding conditions on the landscape. This discrete categorization is common for actual ornamental plants used on residential landscapes. In the model, there are two categories of agent light requirements ($lr \in [1, 2]$), which represent full sun and shade growth requirements. The marigold data generated in the greenhouse was intended to explore the growth response for the full-sun category ($lr = 1$), and the impatiens to represent the shade category ($lr = 2$). In the model, there are three categories for water requirements ($wr \in [1, 2, 3]$) — low, high, and water sensitive. A water-sensitive species experiences a growth reduction if it receives too much water. The marigolds have high water needs ($wr = 2$), and the impatiens are sensitive to overwatering ($wr = 3$). There was no low-water species included in the greenhouse experiments due to time and resource constraints.

Other important properties for plant agents include size (biomass), a shade coefficient, and a water access (wa) parameter. The size parameter influences how much water the agent needs,

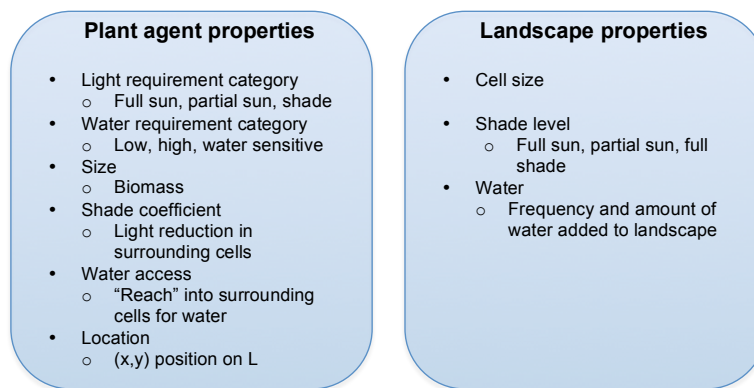


Figure 6.1: Plant-agent and landscape properties. Plant-agent properties determine how the agent responds to, and influences, its surrounding. The landscape properties captures the effects of plant agents on the available light and water resources.

how much water it can access, and how much shade it generates. The shade coefficient and water access parameters determine how an agent influences surrounding cells on the landscape by modifying their light and water properties. The shade coefficient is the light reduction, expressed as a percentage of available light. For example, the shade screen in the greenhouse experiments reduced the light by 70%. Therefore, growth at l_{xy} for p_i assumes the shade coefficient is 0.70. The water-access parameter is the agent's ability to retrieve water from surrounding cells and is also related to the agent's size. Larger values reflect greater competitive ability to extract water from surrounding cells. In this model, the plants all had $wa = 1$.

6.3 Plant growth model — water

Agent responses to watering conditions are drawn from the water-use behavior observed in the greenhouse. If the water that the agent needs is less than the water that is available, the agent experiences a growth reduction. For the *water sensitive* category, agents can be easily overwatered: that is, excess water causes a growth reduction. The relationship between growth (biomass) and water for the marigolds and the impatiens was presented in Chapter 5. These relationships were for a specific set of environmental conditions. To make the model more generally applicable to different environments, an adjustment is made to these equations: the normalized biomass is used instead of absolute biomass of the individual plants. The normalized biomass is defined as $\frac{bio_w}{maxbio_w}$, where $maxbio_w$ is the biomass produced in the most productive watering frequency and bio_w is the observed biomass in a plant's current watering frequency. For the marigolds, a full-sun, high-water species ($lr = 1$, $wr = 1$), the water-response equation for the model is:

$$gr_m = 0.559 \cdot \ln(water_{x,y}) - 0.1991 R^2 = 0.91 \quad (6.4)$$

where $water_{x,y} \in [300, 400, 600, 1200]$ is the mean daily watering rate. For the impatiens, the shade-loving, water-sensitive species ($lr = 3$, $wr = 3$), the water-response equation is piecewise

linear to capture the growth decline that happens when plants are overwatered:

$$\begin{aligned}
 &\text{if } water_{x,y} \leq 600 \\
 &\quad gr_i = 0.0013 \cdot water_{x,y} + 0.233 \\
 &\text{else} \\
 &\quad gr_i = -0.0004 \cdot water_{x,y} + 1.239
 \end{aligned} \tag{6.5}$$

with $water_{x,y} \in [300, 400, 600, 1200]$.

6.4 Plant agents — competition

The water-response curves that describe how plants respond to varying watering frequencies were generated via experiments where plants occupied every cell in the greenhouse trays — no cells were intentionally left blank. An additional experiment in which each plant was surrounded by empty cells confirmed that holding water constant and decreasing competition for water resulted in increased growth. This increase only occurred in water-limited situations — full sun with medium, low, and very low watering frequencies — as shown previously in Figure 5.22. To capture this feature in the model, the fitness score is increased as a function of the number of agents competing for water in surrounding cells for the full-sun, high-water marigolds only. Many of the impatiens were surrounded by several empty cells due to low germination rates in the greenhouse experiments. This additional space could not be correlated with increased size. Therefore, no additional fitness is given to this species. In the greenhouse, the marigolds had a mean of 2.10 competitors in each surrounding cell, and experienced a growth increase of at least 50% in medium, low, and very low watering frequencies in full sun. For the model, this translates to a fitness increase when $water_{x,y} \in [300, 400, 600]$ and $li_{x,y} = 1$. This represents an extremely conservative estimate for the fitness increase in the very low watering frequency conditions. The model also includes a scaled increase, where plants with less than eight competitors in each cell experience some, but not the full 50% increase. This feature is introduced here to explore how various planting densities influence WAE in the fitness function and to introduce heterogeneity in

the effects of spacing for the two species. The fitness increase and the linear scaling are major simplifications of plant behavior. The new water response equation is then:

$$gr'_m = gr_m * (-0.10\mu(waTotal_{x-1:x+1,y-1:y+1} - 1) + 1.8)$$

where

$$waTotal_{x,y} = \sum wa_{x,y} \forall p_i$$

where $waTotal_{x,y}$ is agent competition for water at $l_{x,y}$

(6.6)

6.5 Plant growth model — shade

The other variable that affects growth in this model is shade, which is the level of light reduction from full-sun conditions. The greenhouse experiments confirmed that water and light are not independent: the effect of light on growth is influenced by the amount of water available, and vice versa. A separate growth curve is used in the plant growth model to capture how plants respond in different shade levels in each watering level. In this curve, fitness is represented as a percentage of the maximum mean growth observed in a given watering level. For the full-sun, high-water agents, these relationships are:

$$\begin{aligned} growth_{hw} &= -0.219 \cdot li_{x,y} + 1.232 \\ growth_{mw} &= -0.237 \cdot li_{x,y} + 1.225 \\ growth_{lw} &= -0.046 \cdot li_{x,y} + 1.041 \\ growth_{vlw} &= 0.283 \cdot li_{x,y} + .185 \end{aligned}$$
(6.7)

where $li_{x,y} \in [1, 2, 3]$ represents the level of shade at the plant agent's location. This same approach is used for the shade-loving, water-sensitive agents to yield:

$$\begin{aligned} growth_{hw} &= 0.114 \cdot li_{x,y} + 0.636 \\ growth_{mw} &= 0.162 \cdot li_{x,y} + 0.566 \\ growth_{lw} &= -0.034 \cdot li_{x,y} + 1.025 \\ growth_{vlw} &= 0.091 \cdot li_{x,y} + 0.677 \end{aligned}$$
(6.8)

6.6 Water application efficiency

The previous sections described the factors that determine how well a plant agent can grow in its current light and water conditions; this section describes the other piece of agent fitness — how efficiently plant agents use the water provided. The watering frequency that maximizes growth may not be the same as the watering frequency that produces maximum growth *for the water applied*. In the greenhouse experiments, the marigolds grew best when they were watered every day. However, they produced the most biomass for the water applied when they were watered every other day. This efficiency is termed water application efficiency (WAE); it captures the plant-agent growth in grams per watering applied in mL. WAE is calculated from both the watering frequency and the light at the agent's location. The WAE for the agent is a score that captures efficiency in the agent's current watering frequency relative to the maximum WAE possible. For example, the WAE for marigolds in medium watering conditions (every other day) is 1.0, this is the maximum efficiency possible.

$$WAE_{w,x,y} = \frac{WAE_{wf}}{\max(WAE)}, \forall wf \quad (6.9)$$

where wf is watering frequency (equivalent to the *water* parameter on the landscape), and $WAE_{w,x,y}$ is the WAE in the agent's current *water* conditions. (For example, the most efficient watering frequency will have $WAE_{w,x,y} = 1$.) Next, the output of this equation is modified to include the influence of light within the watering frequency:

$$WAE'_{w,x,y} = WAE_{w,x,y} * g_{w,x,y}(li_{x,y}) \quad (6.10)$$

where $li_{x,y} \in [1, 2, 3]$.

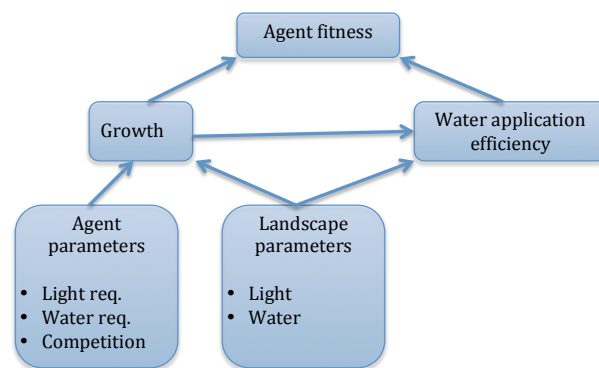


Figure 6.2: Parameters in the growth and WAE components of the fitness function. The growth score captures how well the light and water properties on the landscape match the agent's requirements for light and water. WAE uses the growth score and the properties on the landscape to determine how efficiently the agent used the water provided, i.e. how much did it grow for the water applied.

6.7 Calculating agent fitness

The agent fitness function incorporates the agent's growth equations shown in Equations (6.7) and (6.8), interactions on the landscape due to shade and spacing shown in Equation (6.6), and WAE shown in Equation (6.10). Figure 6.2 shows the two components in agent fitness — growth and WAE. Included in each of these components are the agent and landscape parameters of the model. Growth depends on the agent's light and water requirements, as well as the light and water resources on the landscape. WAE depends only on the agent's growth and the landscape parameters.

The fitness calculation proceeds as follows:

Given a collection of plants, P , on L

$$P = p_i, i = 1 \dots n \quad (6.11)$$

n = number of plants

- **Step 1** — Using the $l_{x,y}$ position of $p_i \forall i$ on L , calculate $li_{xy} \forall x, y$ in L .
- **Step 2** — Determine $waTotal_{x,y} \forall x, y$ using $l_{x,y}$ position of p_i and $wa_i \forall i$
- **Step 3** — Calculate the growth response to water for p_i using the appropriate water-response curves:

$$f_w = 0.559 \cdot \ln(water_{x,y}) - 2.89 \quad (6.12)$$

for the high-water species ($wr = 2$), and:

$$\begin{aligned} &\text{if } water_{x,y} \leq 600 \\ &f_w = 0.0013 \cdot water_{x,y} + 0.233 \\ &\text{else} \\ &f_w = -0.0004 \cdot water_{x,y} + 1.239 \end{aligned} \quad (6.13)$$

for the water-sensitive species ($wr = 3$). Here, $water_{x,y} \in [300, 400, 600, 1200]$.

- **Step 4** — Calculate additional growth effects due to the competition for surrounding water resources calculated in Step 2:

$$f'_w = f_w * g(neighbors, water_{x,y}, li_{x,y}) \quad (6.14)$$

where $g(neighbors, water_{x,y}, li_{x,y})$ is a function of the number of neighbors of a plant agent, $water_{x,y} \in [300, 400, 600, 1200]$ is its watering frequency, $li_{x,y} \in [1, 2, 3]$ is the shade level, as described by Equation (6.6). In the greenhouse experiments, a growth increase was observed when $water_{x,y} \in [300, 400, 600]$.

- **Step 5** — Include growth response to light at the given water level:

$$f_i = f'_w * g_w(li_{xy}) \quad (6.15)$$

for $p_i \forall i$.

- **Step 6** — Calculate the WAE score using:

$$WAE_i = \frac{WAE_{w_{xy}}}{\max WAE_{w_{xy}}} * g_{w_{xy}}(li_{xy}) \quad (6.16)$$

- **Step 7** — Calculate the final fitness score including growth and WAE components:

$$fitness_i = \alpha \cdot f_i(water_{xy}, li_{xy}) + \beta \cdot WAE_i \quad (6.17)$$

where α and β are user-selected weights to establish the priority of each term. When $\alpha > \beta$, the priority is on growth, and vice versa, the priority is on WAE. These weights address the conflicting priorities of growth and water use. The tradeoff between them and how they might be tuned is discussed in more detail in Chapter 8.

- **Step 8** — Calculate the final system fitness score for the arrangement:

$$landscapeFitness = \sum_{i=1}^n fitness_i \quad (6.18)$$

To evaluate how well the model reflects the actual biomass production, the fitness scores and mean biomass for both the marigolds and the impatiens were compared for each light and water combination. This comparison was performed for conditions where plants had neighbors and where they were surrounded by empty cells. Table 6.1 shows the fitness score, mean biomass, water and light levels for the marigold model. The same information is provided for the impatiens in Table 6.2. The final column in Table 6.1 shows whether the plant had empty space around it. The fitness scores in these tables represent the condition where $\alpha = 1.0$ and $\beta = 0.0$ in Step 7 of the fitness function — i.e. when WAE is not included in the calculation. Generally speaking, the fitness scores calculated from the model do accurately reflect the mean biomass of the plants in different conditions. For the marigolds, the Pearson’s correlation coefficient between the fitness scores and the mean biomass was 0.87, with a p-value of $2.3e^{-05}$. For the impatiens, the correlation was 0.91 with a p-value of $2.6e^{-05}$. These relationships are shown visually in Figure 6.3.

The linear fits shown in the image are described by:

$$\begin{aligned} fit_{mari} &= 12.8480 \cdot biomass - 0.0278, R^2 = 0.76 \\ fit_{imp} &= 49.60 \cdot biomass + 0.1167, R^2 = 0.84 \end{aligned} \tag{6.19}$$

This chapter showed how the greenhouse experiments described in Chapter 5 were used to build a model of plant growth and interactions on a landscape. In this model, each plant agent has light and water requirements that can be used in a fitness function, which determines how well the agent’s requirements match the available resources. In the next chapter, a novel method for optimizing plant-agent locations is introduced.

Table 6.1: Fitness scores for marigold light, water, and spacing combinations. These scores are calculated using the model equations described in this chapter. The biomass values are from the greenhouse experiments described in Chapter 5. The Shade and Water columns are the light reduction and water inputs to the fitness equations, where $shade \in [1, 2, 3]$ is full sun, partial sun, and shade respectively, and $water \in [1200, 600, 400, 300]$ is high, medium, low, and very low respectively. The Space column describes whether plants were grown with all surrounding cells populated by other plants or by empty cells.

Fitness score	Mean biomass	Water	Shade	Space(Y=1, N=0)
0.1397	0.0112	Very low	Full sun	0
0.2095	0.0535	Very low	Full sun	1
0.2241	0.0211	Very low	Partial	0
0.3086	0.0258	Very low	Shade	0
0.3525	0.0331	Medium	Shade	0
0.4147	0.0308	Low	Shade	0
0.4358	0.0318	Low	Partial	0
0.4569	0.0339	Low	Sun	0
0.5151	0.0457	Medium	Partial	0
0.6172	0.0417	High	Shade	0
0.6777	0.0628	Medium	Sun	0
0.6854	0.0498	Low	Sun	1
0.8522	0.0608	High	Partial	0
1.0165	0.0785	Medium	Sun	1
1.0873	0.0741	High	Sun	0

Table 6.2: Fitness scores for the impatiens in the light and water combinations. Unlike the marigolds, additional spacing around each plant does not affect the fitness scores for this species.

Fitness score	Mean biomass	Water	Shade
0.4785	0.0088	Very low	Sun
0.5352	0.0081	Very low	Partial
0.5693	0.0105	High	Sun
0.5918	0.0107	Very low	Shade
0.6558	0.0112	High	Partial
0.6950	0.0104	Low	Shade
0.7206	0.0105	Low	Partial
0.7375	0.0119	Medium	Sun
0.7423	0.0136	High	Shade
0.7462	0.0112	Low	Sun
0.9016	0.0174	Medium	Partial
1.0657	0.0176	Medium	Shade

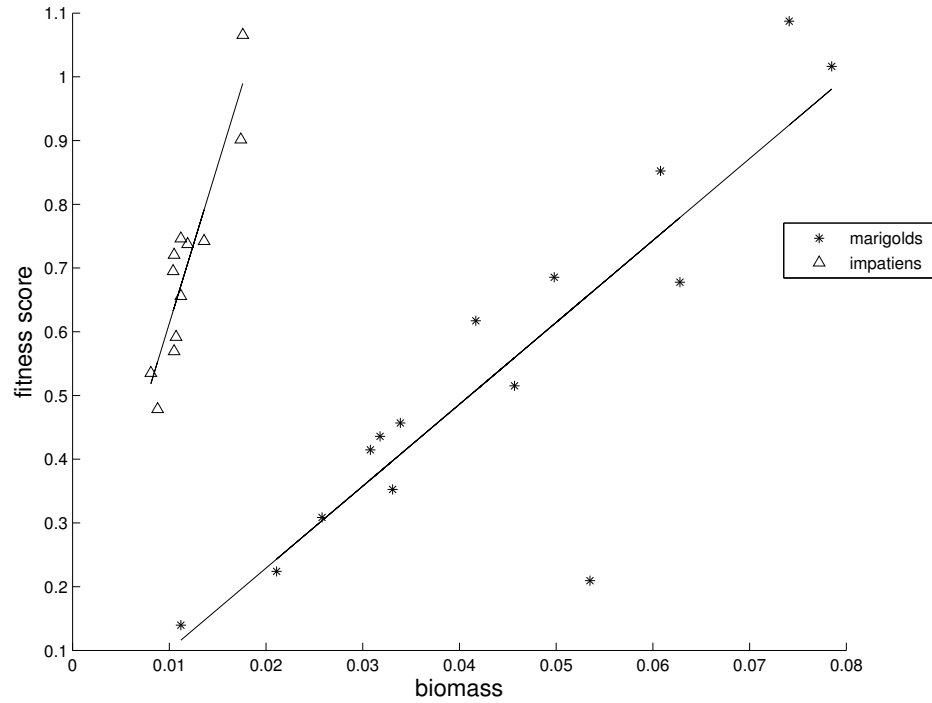


Figure 6.3: Correlation between the fitness scores in the model and biomass for marigolds and impatiens in the greenhouse experiments. The linear fits shown here have $R^2 = 0.76$ for the marigolds and $R^2 = 0.84$ for the impatiens. There is one outlier for the marigolds, which is the fitness score assigned to the plants grown in full-sun, very low watering conditions with surrounding empty cells. The model uses a conservative estimate for these conditions when plants have reduced competition from nearby plants. This conservative estimate means that the model underestimates fitness in these conditions.

Chapter 7

An agent-based search algorithm

Examples of search algorithms that are typically used on location problems were presented in Chapter 3. This thesis presents a non-standard location problem and a search algorithm designed around the problem's unique features. This algorithm is designed to mimic how individual plant communities evolve over time as individual plants respond to their local resource conditions. In this paradigm, each plant is concerned only with its own survival, not with how its location affects the community as a whole — or, in computational terms, the global fitness. The level of global control over solution fitness is the primary difference between this optimization algorithm and those previously mentioned. This algorithm calculates the fitness of the individual items (plant agents) on each iteration and only calculates global fitness when all plant agents have selected a location. This process repeats, with a global fitness calculation following individual agent moves, until there is no improvement in the global fitness score. Reducing the global fitness calculations also reduces the computational complexity of the search, requiring an $O(1)$ operation to calculate individual fitness instead of the $O(n)$ operation required to calculate the fitness of all items.

The agent-based search routine combines local search and random jumps. Each agent first searches nearby locations to improve its fitness score. If it does not find a better location, the agent jumps to a random location on the landscape. There are a fixed number of locations that an agent can sample either through local moves or random jumps. The agent's objective with each move is to improve upon its current fitness score. There are three fundamental parameters in the agent-based search routine:

- **threshold** — the minimum fitness score that a plant agent attempts to improve upon during the search.
- **moves allowed** — the number of times a plant agent can try out new locations.
- **local neighborhood** — the maximum number of cells in each direction that a plant agent can search before making a random jump.

The search proceeds as shown in Algorithm 5. A collection of plant agents is defined by the user, where each plant agent has requirements for light and water. First the landscape is initialized in lines 1-6, which includes randomly assigning initial locations for all plant agents, setting the light and water properties for each cell, and setting the *threshold* to a uniform initial value. Two iterators control the execution of the algorithm. There is an outer loop controlled by the variable t , which controls the value of the *threshold* parameter for each agent upon entry to the search algorithm at iteration t . There is also an inner loop that uses the *threshold* to control execution at the individual agent level within the algorithm.

The algorithm loops through the agents serially, updating the light on the landscape (viz., line 11) using the agents' current positions, and then updating their fitness scores (viz., line 12) using the fitness function described in Chapter 6. If an agent's fitness score is above its current *threshold*, the agent stays at its current position, and the algorithm continues to the next agent (viz., lines 13-14). Otherwise, if the agent has moves available, i.e. it has not yet exhausted the number of locations it can "try out", the algorithm first searches locally within the *local neighborhood*, which is a parameter of the algorithm. Recall that one of the distinguishing features of this location problem is influence of plants upon their local light and water resources. The local search step in the optimization algorithm is included to respond to conditions where a small adjustment in the agent's location could move it into or out of the influence of a nearby agent, thereby changing its fitness. In this local search phase, agents start with a small step in all directions, then increase their search radius if no fitness-improving locations are found, up to the *local neighborhood* radius (viz., line 17). If the agent finds a better location in this local search process, and that location is

Algorithm 5 Agent-based search algorithm

```

1: Given a collection of plant agents,  $P = p_i, i = 1 \dots n$ 
2: on landscape  $L, l_{x,y}, x = 1 \dots rows, y = 1 \dots columns$ 
3:  $light_{x,y}$  is the light available at location  $x, y$ ;  $wl_{x,y}$  is the water available at location  $x, y$ 
4: Assign unique, random  $l_{x,y} \forall p_i$ ;  $l_{x,y} = 1$  if occupied,  $l_{x,y} = 0$  o.w.
5: Set  $t = 0$ 
6: Set uniform  $threshold_{i,t} \forall i$ 
7: repeat
8:   Store initial  $l_{x,y}, \forall p_i$ 
9:   repeat
10:    for all  $p_i$  do
11:      Update  $light_{x,y}, \forall x, y$  using location of  $p_i \forall i$ 
12:      Calculate  $fitness_i$  for  $p_i$  at  $l_{x,y}$ 
13:      if  $fitness_i > threshold_{i,t}$  then
14:        Leave  $p_i$  at  $l_{x,y}$ 
15:      else
16:        if  $p_i.moves < moves\ allowed$  then
17:          Search  $l_{x-ln:x+ln, y-ln:y+ln}$  for  $fitness'_i$  at  $l'_{(x,y)} > fitness_i$  at  $l_{x,y}$ 
18:          if  $fitness'_i > fitness_i$  then
19:            if  $l'_{(x,y)} = 0$  then
20:              Move  $p_i$  to  $l'_{(x,y)}$ 
21:            else
22:              Calculate  $fitness_j$  for  $p_j$  already at  $l'_{(x,y)}$ 
23:              if  $fitness_i > fitness_j$  then
24:                Move  $p_i$  to  $l'_{(x,y)}$ 
25:                Select random  $l_{x,y}$  for  $p_j$ , where  $l_{x,y} = 0$ 
26:              end if
27:            end if
28:          else
29:            Select random  $l_{x,y}$  for  $p_i$ , where  $l_{x,y} = 0$ 
30:          end if
31:          Increment  $p_i.moves$ 
32:        end if
33:      end if
34:    end for
35:    until  $fitness_i > threshold_{i,t}$ , or  $p_i.moves \geq moves\ allowed, \forall i$ 
36:    if  $fitness_i < threshold_{i,t}$  then
37:      Move  $p_i$  back to initial  $l_{x,y}$ , if  $l_{x,y} = 0$ 
38:    end if
39:    Calculate  $fitness_i, \forall i$ 
40:    if  $count(fitness_i > threshold_{i,t} \forall i) > count(fitness_i < threshold_{i,t} \forall i)$  then
41:      Set  $threshold_{i,t+1} = fitness_i \forall i$  where  $fitness_i > threshold_{i,t}$ 
42:      Increment  $t$ 
43:    else
44:       $t = 0$ 
45:    end if
46:  until  $t = 0$ 

```

currently unoccupied, it moves to that location (viz., lines 18-20). The fitness calculation during this phase is for the individual agent only — not all agents on the landscape. As a result, the consequences of each agent’s move upon other agents are unknown.

It is possible that the agent could find a fitness-improving location in the local search, but that location could be occupied by another agent. In this case, the agents “compete” for the spot. This process mimics how real plants compete for locations: the species best adapted for the resources survives, and the other dies. Algorithmically, the fitness scores for the competing plant agents are calculated and the agent with the highest fitness score gets the spot, and the other agent gets a random, unoccupied location on the landscape (viz., lines 22-25).

If an agent does not find a better location via the local search process before reaching its maximum search radius, it jumps to a random, unoccupied location on the landscape (viz., line 29). This random jump does not first check for the effects on the fitness score, either for the agent, or the entire agent population. Here, this algorithm differs from the standard algorithms for location problems. From the perspective of mimicking an ecological process, this random jump simulates seed dispersal, where seeds are cast out and those that land in favorable conditions have a higher probability of germinating than those that land in less favorable conditions. Following the local move or the random jump, the number of moves that the agent has made in the search process is incremented by 1 (viz., line 31).

After looping through all agents, the search begins again at the first agent in the list ¹ and repeats either until all agents have achieved fitness scores above the *threshold* or no agent has any more moves available (viz., line 35). Requiring an improvement in fitness over the *threshold* on each iteration of the algorithm becomes problematic if the *threshold* is at the highest fitness score that an agent can achieve on a given landscape. When this happens, agents end up being bumped from this ideal location (viz., line 12), and search futilely for a better location. To correct this problem, the algorithm includes an “undo” feature, which checks fitness at the end of this inner loop and returns agents to their initial location if they did not improve upon their initial fitness

¹ Experiments described in Chapter 8, but this is not a requirement for the algorithm

(viz., lines 36-38). This is not a perfect solution, as another agent could have moved in during that step to occupy the location, or the conditions at the location could have changed such that it is no longer desirable. Following the “undo” movements, there is a final fitness calculation for all agents, to capture the conditions where agents moved back to their initial locations, and this inner search loop exits.

To determine whether this inner execution loop actually improved fitness scores for the individual agents, the algorithm compares the number of agents with improved fitness scores to the number with decreased fitness (viz., lines 40-42). If the former exceeds the latter, the algorithm resets the *threshold* to match each agent’s current fitness score for the next iteration, increments the iterator t , and re-enters the inner search loop. If the number of agents with a decrease in fitness exceeds the number with an increase in fitness, the search exits.

Chapter 8

Experimental methods

This thesis presents a novel approach for optimizing plant placements on a landscape so as to conserve water in residential systems. This chapter describes how the approach presented in this thesis was evaluated via both numerical simulations and a validation experiment in the greenhouse with live plants. The numerical simulations investigated:

- How optimization strategies with different fundamental characteristics performed on this problem, in terms of both the fitness scores of the results and the run-time characteristics of each algorithm.
- How effectively the agent-based model captured the heterogeneous growth requirements of the individual agents, each agent's affect on the water resources on the landscape, and the local interactions between agents.
- How parameters in the fitness function influenced individual agents' growth and WAE scores in different watering frequencies.
- How individual agents' locations in different watering frequencies affected the emergent property of the total water applied to the landscape.

In addition to these numerical simulations, an experiment was performed with live plants in the greenhouse. Two arrangements were generated numerically — one by placing agents at random locations and one by optimizing placements for water application efficiency — and then selected

for planting. This experiment was used to demonstrate that model predictions about growth and water use could work in practice with real plants.

8.1 Optimization strategies

Genetic algorithms (GA) and simulated annealing (SA) engage different approaches for exploring the search space. The GA (described in Chapter 3, Algorithms 2 and 3) combines segments of individual solutions in a population to generate new solutions to improve fitness. In contrast, SA (described in Chapter 3, Algorithm 4) seeks to modify individual elements in a single solution to improve fitness. The fitness calculation is global in both algorithms — all elements are evaluated after each iteration to determine if solution fitness has improved.

In the agent-based routine designed as part of this work covered in Chapter 7, individual agents react to their immediate surroundings and move around on the landscape to improve their individual fitness. Global fitness is measured after the locations for all agents have stabilized, at which time the global fitness calculation counts the number of individuals that improved their fitness scores on the previous iterations and compares that number to the number whose fitness scores decreased.

All three algorithms have tunable parameters that influence how the algorithm proceeds, including the runtime of the algorithm and the percentage of the search space evaluated. The parameters for each algorithm include:

Genetic algorithms:

- **Population size** — Number of candidate solutions from which to select parents for the next generation of candidate solutions.
- **Parent selection algorithm** — Method for selecting parent solutions for the crossover process that generates new candidate solutions.
- **Mutation rate** — Frequency at which a random individual is selected from the population and its value is modified.

- **Crossover point** — Position on a parent solution that controls how much of a new candidate solution comes from each parent.

Simulated annealing:

- **Cooling rate** — Rate at which the probability of accepting a non-improving solution, i.e. a solution that decreases the global fitness, decreases during the search process.
- **Initial temperature** — Starting temperature that, combined with the cooling rate, determines the number of iterations for the algorithm.

Agent-based search:

- **Threshold** — Initial fitness score for each agent that the agent tries to improve upon during the optimization process.
- **Local neighborhood** — Local search radius that an agent explores for fitness score improvements before making a random jump.
- **Moves allowed** — Number of locations that an agent can “try out”, either through local moves or random jumps.

The number of parameters in these algorithms and the potential combinations of these parameters make it difficult to do a complete evaluation of each algorithm, or claim that the evaluation performed here is a fair comparison between the algorithms. One algorithm may outperform another, but this performance is specific to the parameters explored. In this evaluation, the range of values for each parameter was established using values observed in the literature and preliminary tests that showed that values outside this range did not improve the results. It is possible that the value range includes a local optimum for each algorithm. However, it is impossible to know if this is the case without performing an extensive evaluation of all possible parameter combinations, a task that is outside the scope of this work.

For SA, the evaluation included modifying the T and α parameters in the $T = \alpha \cdot T$ equation on line 14 of Algorithm 4. Cooling rates (α) were evaluated between 0.99 and 0.50. A range of initial temperatures (T) was explored; this value was set to 15.0 after initial tests determined that values above this number did not increase fitness scores of the results. For the GA, the modified parameters were the mutation rate and the number of iterations for the algorithm. Mutation rates between 0 and 20 individual modifications on each iteration were used. For all GA evaluations, parents were selected randomly and the crossover point was set at the midpoint of those candidate solutions. More complicated techniques exist for selecting parents and crossover points, such as only crossing parents with the highest individual fitness scores and using multiple crossover points. However, this thesis represents the first time that a GA has been evaluated on this problem. Therefore, the simple approach was used. For the agent-based search, the local neighborhood was varied between 0 and 3 cells, and individual agents were allowed between 1 and 15 moves. The initial fitness threshold was set at 0.2, which is a fitness score that represents a barely surviving plant according to the greenhouse model.

Each of these algorithms includes elements of randomness, from a random initial configuration used in all three algorithms to random moves in the agent-search and SA and random mutations and parent selection in the GA. This randomness generates stochastic results, so none of the algorithms will necessarily produce the same answer each time it runs. Evaluating the success of an algorithm, therefore, requires multiple executions to determine how well it performs not just one time as a result of a good starting configuration, but on average. To accomplish this, 100 iterations of each simulation were performed for each algorithm. These data were analyzed by computing the mean fitness scores, watering frequency for individual plant agents and the total water applied over the entire landscape, how algorithm parameters affected results, and the execution time of each algorithm on a Janus 16,416-core supercomputer¹.

The GA, SA, and agent-search routines were written in C by the thesis author. In addition to

¹ This work utilized the Janus supercomputer, which is supported by the National Science Foundation (award number CNS-0821794) and the University of Colorado Boulder. The Janus supercomputer is a joint effort of the University of Colorado, Boulder, the University of Colorado, Denver and the National Center for Atmospheric Research.

these implementations, the GA provided in the Matlab Global Optimization Toolbox was applied to this problem. In the following sections, these results are discussed and compared, along with the benefits and limitations of the commercial optimization package.

8.2 Landscape conditions and agents

The numerical simulations and the greenhouse validation experiment used a landscape with the same light and water conditions to provide consistency between these different experiments. This approach was intended to validate the claim that results generated through simulation can be reproduced with live plant experiments. The landscape size was 24x9 cells, the same size as the grid in the greenhouse experiments described in Chapter 5. To generate variable light on the greenhouse landscape (simulating the presence of a shade tree), two simulated shade screens were placed at different positions. A 4-inch shade screen was placed at row 20, columns 1 through 4, and a 5-inch shade screen, was placed at row 1, columns 3 through 7. This arrangement and the method for calculating shading that is described in Chapter 5 generated the profile of sun, partial sun, and shade shown in Figure 8.1 that was used in the numerical simulations. In this image, the blue areas are shade ($li = 3$), the green areas are partial sun ($li = 2$), and the red areas are full sun ($li = 1$).

The four watering frequencies used in the greenhouse experiments were also used in this simulated landscape, as shown in Figure 8.2. The high watering frequency region was watered every day; the medium region, every other day; the low region, every third day; and the very low region, every fourth day. The total amount of water applied to the entire landscape was calculated from the number of occupied cells, the watering frequency for the cell and surrounding cells, and the simulation time in days. In all simulations, the water applied for an arrangement was calculated

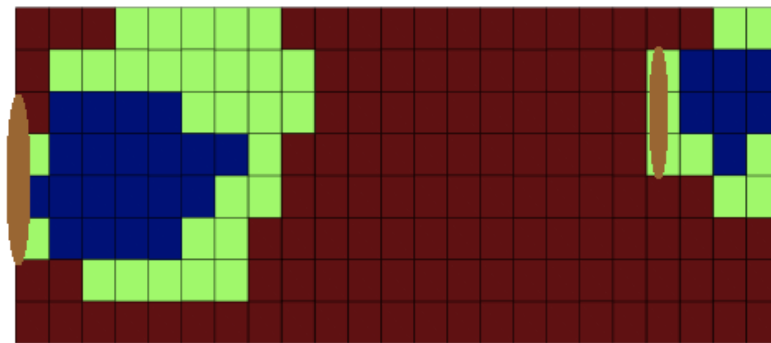


Figure 8.1: Light distribution of shade, partial sun, and full sun conditions for simulations and live validation experiment. Blue areas are shade, green is partial sun, and red is full sun. The brown ovals show the positions of the simulated shade screens.

as:

Given a collection of plants, P on landscape L

Watering radius of p_i at $l_{x,y}$ is $l_{x-1:x+1,y-1:y+1} \forall p_i$

$$w_{x,y} = \begin{cases} 1 & \text{if } l_{x,y} \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (8.1)$$

$$waterApp = \sum_x \sum_y w_{x,y} * w * dw$$

where $dw \in [1...nDays]$ is the number of days where $l_{x,y}$ is watered, and $nDays$ is the number of days in the simulation. Using the watering frequencies in the greenhouse experiments, w was $\frac{1200 \text{ mL}}{216 \text{ cells}} = 5.50 \text{ mL}$, where 1200 mL was the water applied to an entire tray with each watering, and 216 was the number of cells in a tray. Based on this calculation, the upper bound on the amount of water applied to each watering region in these simulations was 320 mL daily to the high watering frequency region, 280 mL every other day to the medium watering frequency region, 320 mL every three days to the low watering frequency region, and 280 mL every four days to the very low watering frequency region, for a maximum total of 7640 mL every 12 days. However, the *actual* amounts depended upon the positions of the plants. If multiple plants had an overlapping watering radius it reduced the water applied to the landscape.

These simulations used two types of plant agents — the full-sun, high-water agents to simulate the marigolds and the shade, water-sensitive agents to simulate the impatiens. The performance of varying numbers of these two species was evaluated on the simulated landscape to capture how the fitness scores of each species changed as competition increased for their particular growing conditions and for the water on the landscape. The least-constrained simulation used 20 plant agents, divided evenly between both species. In this simulation, there were sufficient locations on the landscape for both species. This was intended to evaluate the optimization strategies on a problem where agents were not competing for space on the landscape. The greatest number of plant agents used was 100, including 50 marigolds and 50 impatiens. In this simulation, a plant agent occupied every other cell on the landscape and there was competition not only for growing conditions but also for water. This simulation was intended to evaluate how each optimization strategy

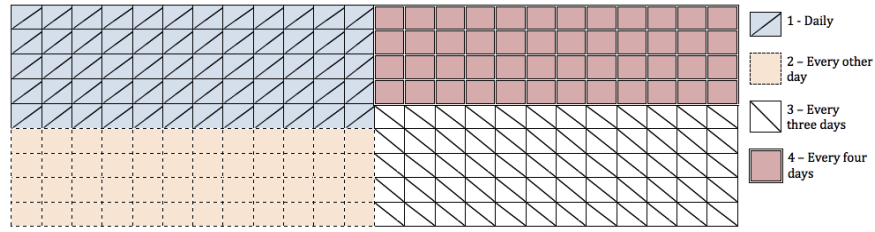


Figure 8.2: Watering frequencies on the landscape used for all numerical simulations as well as the live-plant validation experiment. The landscape includes all watering frequencies used in the greenhouse experiments; 1200 mL of water was applied at each watering.

performed under constrained conditions where there were few good locations for each plant agent. Additional simulations were performed with populations evenly divided between the two species using 30, 50, 60, 70, 80, or 90 plant agents. There was also one plant agent collection with 30 marigolds and 15 impatiens that was used for both the simulations and the greenhouse validation experiment. This collection was created with the validation experiment in mind; the rationale for selecting this number of agents is described in Section 8.4 later in this chapter.

8.3 Water application efficiency

The multiobjective fitness function presented in this thesis includes two pieces:

- A growth score to represent how well the light and water conditions at the plant agent's location match its light and water requirements
- A water application efficiency (WAE) score to capture how much the plant agent grew for the water it received.

Both the *growth* and the *WAE* calculation for the agents were described in Chapter 6. There are two user-defined parameters in the fitness function, α and β , that control the weight given to each objective during the optimization:

$$f = \alpha \cdot growth + \beta \cdot WAE \quad (8.2)$$

When $\alpha > \beta$, *growth* is weighted more heavily than *WAE*, and vice versa. The elements of *growth* and *WAE* are potentially conflicting: plants need water to grow, and reducing water could also reduce growth. This interaction suggests that a solution that maximizes *WAE* might do so at the expense of *growth*, and vice versa.

Simulations evaluated how the α and β values influenced *WAE* in the fitness calculation, as well as the mean *WAE* for the entire arrangement, where:

$$WAE_{arr} = \frac{1}{n} \sum_i^n WAE_i \quad i = 1 \dots \text{number of plants} \quad (8.3)$$

These parameters' influence on the total water applied to the landscape was calculated using Equation (8.1). Values for α and β ranged between 0.0 and 1.0 in increments of 0.1, where $\alpha + \beta = 1.0$, $\forall \alpha, \beta$. For example, a simulation could use $\alpha = 0.9$ and $\beta = 0.1$. Both the *WAE* and *growth* scores have a range of 0.0 to 1.0, and setting the α and β values to be offsetting and in the same range maintains this fitness score range.

The fitness function was evaluated on all plant agent collections described in Section 8.2 to evaluate not only the effect of α and β on individual fitness and watering frequency, but also how different numbers of plants and α and β values influenced plant spacing, competition, and total water applied to the landscape.

8.4 Validating a simulation in the greenhouse

One of the claims of this thesis is that solutions produced in simulation can be validated with live plants. This validation means that a plant arrangement predicted to grow and use less water than another arrangement will actually do so when live plants are placed in soil in the predicted configuration, and the necessary water is applied to the arrangement.

The validation experiment performed here used two arrangements: one randomly generated and one generated through the optimization process with the priority placed on *WAE* in the fitness score calculation. To select a random arrangement for this experiment, 100 arrangements were generated using $\beta = 0.6$ in the fitness calculation. The arrangement that produced the lowest

fitness score was selected for planting. Optimized arrangements were generated using the same β value and number of simulations. The arrangement with the highest fitness score was also planted.

Working with live plants presents challenges that do not arise in simulations. For this experiment, the challenges were germinating enough plants with similar sizes to provide consistent starting conditions, and germinating two species with different growth patterns at the same time. Growing the plants from seed in their selected locations in the random and optimized arrangements could give one species an advantage and generate unforeseen conditions on the landscape. For example, by the time the impatiens germinated, the marigolds could already be so large that they shade nearby cells and out-compete neighboring plants for water. To address this difference in germination, plants were germinated at different times. Impatiens were planted on February 8, 2012, and the marigolds were planted on February 20, 2012. Plants were then transplanted into their respective locations. Plants were watered daily for three day after transplanting, at which time it was clear that they had all survived the transplant process.

The other challenge in this experiment was assuring that all plants were a consistent size at the beginning of the experiment. In each greenhouse tray, there were 45 plants, which included 30 marigolds and 15 impatiens. This was the number of plants of each species that appeared to be of similar size at the beginning of the experiment.

The experiment was planned for three weeks, the same duration as the greenhouse experiments described in Chapter 5. Each day, water was applied to the plants in the four watering regions (high water, every day; medium water, every other day; low water, every third day; very low water, every fourth day). The configuration of plants in the random arrangement is shown in Figure 8.3, and the optimized arrangement is shown in Figure 8.4. The amount of water applied at each watering was calculated using Equation (8.1). In the random arrangement, 305 mL of water was applied to the high-water region daily, 220 mL was applied to the medium-water region every other day, 235 mL was applied to the low-water region every three days, and 220 mL was applied to the very low region every four days. Over the duration of the experiment, a total of 8,780 mL of water was applied to the random arrangement. For the optimized arrangement, 90 mL was applied

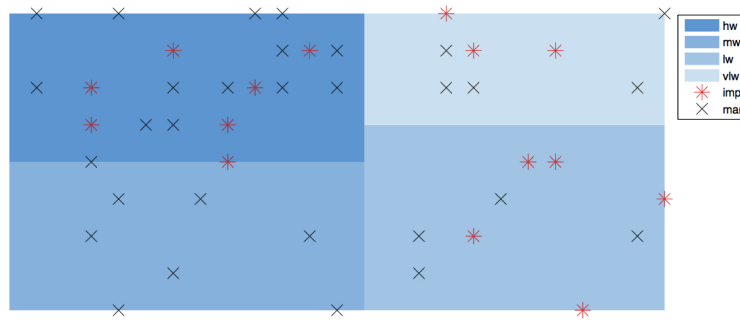


Figure 8.3: Random arrangement used for greenhouse validation experiment. Plants were evenly distributed over the landscape, and each watering frequency contained a mix of each species. This arrangement was used for comparison to an arrangement of the same collection of plants that was optimized by agent-search for water application efficiency.

daily to the high-water region, 250 mL was applied to the medium-water region, 320 mL was applied every three days to the low-water region, and 170 mL was applied every four days to the very low-water region. A total of 5,560 mL of water was applied to this arrangement, approximately 63% of the amount applied to the random arrangement. At the end of the experiment, plants were dried and weighed. The biomass and water applied for each arrangement, as well as the correlation between these values and the fitness scores, were calculated.

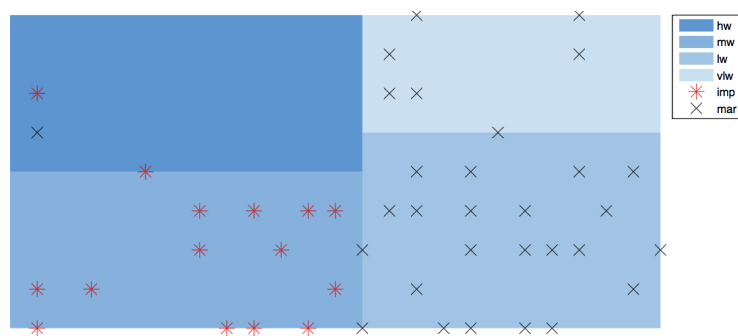


Figure 8.4: Arrangement optimized for water application efficiency used in the greenhouse validation experiment. This arrangement required 37% less water than the random arrangement of the same plants that is shown in Figure 8.3. In this arrangement, there were very few plants in the high watering frequency region: most plants for both species were in either medium or low watering frequency. This placement reflects the emphasis on water efficiency over growth in the fitness function. There were also only a few plants in the very low watering frequency region, which reflects the other trade-off in this problem — plants need water to grow.

Chapter 9

Results

This chapter discusses the findings of the simulations and live-plant validation experiment that are described in the previous chapter. The analyses can be divided into two broad categories: evaluating the optimization strategies and evaluating the model and fitness function. However, these categories are not independent. The optimization strategies maximize plant-agent fitness using the model of how the agents perform in different light and water conditions.

All three optimization strategies — SA, GA, and agent search — are designed to maximize fitness scores for a collection of plant agents on a landscape. To evaluate these strategies, the fitness scores that each strategy produced on the simulated landscape are compared. All plant collections described in the previous chapter are used: 20 to 100 plant agents evenly divided between species, and 45 plant agents with 30 marigolds and 15 impatiens. This comparison provides information on how the strategies performed on problems with different constraints for space, growing conditions, and water. A significant factor in how these strategies perform on any optimization problem is the parameter values selected for each strategy, so an important component of this evaluation is an exploration of the associated parameter space and an assessment of the effects upon fitness and execution time.

To evaluate the model and fitness function, the fitness scores for plant agents in various light and water conditions are compared. Plant agents in their desired light and water conditions should also have high fitness scores, and plant agents in less desirable conditions should have lower fitness scores. The desired conditions depend upon the α and β parameters in the fitness function

that place priority on either *growth* or water application efficiency (*WAE*). The effects of the α and β parameters on the watering frequency of individual plants as well as the emergent property of the total water applied to the landscape are compared as part of the fitness-function evaluation.

One of the claims of this dissertation is that the predictions of this agent-based model can be validated with live plants in a greenhouse. The live-plant validation study was designed to validate this claim; its results show that the optimized arrangement does indeed use less water.

9.1 Fitness scores

Figure 9.1 shows the best fitness scores for each of the three optimization strategies for each collection of plant agents. The x-axis in this figure is the number of plants used in the simulation and the y-axis is the mean fitness score for all plants on the landscape in a simulation. Each data point on this graph is the mean of 1100 simulations including 100 simulations for each α, β parameter combination, where $\alpha \in [0...1.0]$, and $\beta \in [0...1.0]$, $\alpha + \beta = 1.0$, $\forall \alpha, \beta$. These results are for a cooling parameter value of 0.99 for SA, 1700 iterations and one mutation for the GA, and 11 moves allowed and 3-cell local neighborhood for the agent search: the values that produced the highest fitness scores for each strategy. SA produced the highest fitness scores of the three strategies and the difference in the fitness scores between SA and the other two strategies increased as more plant agents were added to the landscape. However, even with 100 plant agents, the difference in these scores was less than 10%. The scores for agent search and the GA were comparable. However, when the runtime of each algorithm is considered (discussed later in this chapter), agent search outperformed the GA.

For all three strategies, fitness scores dropped as more plant agents were added to the landscape due to competition for water and the limited availability of locations with light and water resources that matched the agents light and water requirements. Recall that the impatiens did not grow well in high-water, full-sun conditions in the greenhouse. Yet, these were the predominant conditions on the landscape introduced in the previous chapter — most locations were either full sun, high water, or both, as shown in the shade and water maps in Figures 8.2 and 8.1. In these

simulations, even with only 20 plants, the impatiens plant agents were in full shade or partial sun conditions only 60% of the time in the solutions for all three optimization strategies. This number dropped to 50% for 30 plant agents, and dropped further to 34% when there were 60 or more plant agents on the landscape. For the marigold plant agents, there were sufficient locations that matched this species' growth requirements, even when the landscape was heavily populated. With 100 plant agents on the landscape, only 4% of the marigold agents were not in full sun conditions. For all other plant agent collections, where there were fewer than 100 agents, the marigold agents were in full sun even more frequently.

For the marigold agents, the reduction in fitness scores as the landscape became more populated was due to increased competition for water. One of the components of the agent-based model is the increased fitness for this species when agents are surrounded by vacant cells in lower watering frequencies. This additional space provides additional access to water, reducing water stress and resulting in higher fitness scores. With more plant agents on the landscape, there were few empty cells surrounding each agent. According to the model, if the mean number of competitors in all surrounding cells is greater than three (calculated using Equation (6.6)) in full-sun, reduced watering frequency conditions (a value determined from the greenhouse experiments), the agent experiences a reduction in growth. Figure 9.2 shows the percentage of the marigold plant agents in the SA solutions with more than three neighbor competitors as a function of the number of plants on the landscape. The behavior shown in this figure was characteristic of all three optimization strategies. With 60 plants on the landscape, about 10% of agents in the optimized arrangements had three or more neighbors and experienced a fitness reduction due to this competition. Increasing the number of plants also increased competition. With 80 or more plant agents on the landscape, almost 100% of the plants in the optimized arrangements experienced a growth reduction due to competition. These competition effects were meaningful not only for individual plant-agent fitness, but also for how this individual spacing influenced the total water applied to the landscape, which is discussed later in this chapter.

9.1.1 Evaluating individual plant-agent mortality

The fitness scores presented in Figure 9.1 are the mean optimized values for all plant agents in a simulation. However, given the domain being considered, the performance of individual plants, and in particular, how many individual plants “survived”, is also important. In the greenhouse experiments described in Chapter 5, plants in densely populated conditions in the low and very low watering frequencies did not perform well: they died or were severely stressed. The fitness function calculation assigned plants in these conditions a fitness score of 0.45 or lower. Therefore, 0.45 was used in the numerical simulations here as the survival threshold to evaluate how many plant agents “survived”: plant agents with fitness scores below this fitness threshold were considered to be dead.

Figure 9.1 shows that the mean fitness scores of the optimized solutions decreased for all three strategies as more plant agents were added to the landscape. However, these reduced fitness scores did not result in increased mortality on the landscape. The survival rate for all plant agents was above 98% for all plant populations and for all optimization strategies.

One possible explanation for the low mortality rate is that optimization was not needed on this landscape to prevent mortality for these particular plants, i.e. just adding plants randomly to the landscape would produce solutions with mortality rates similar to the optimized solutions. However, additional simulations using the same collections, where plant agents were added randomly to the landscape and their fitness scores were calculated, showed that the fitness scores for random placements were lower than the fitness scores from any of the three optimization algorithms. In addition, several of the plant agents had fitness scores below the 0.45 survival threshold. Figure 9.3 shows the mean fitness scores for all simulations generated with this random-placement approach. The mortality rate was consistently around 13% for all plant collections. The performance of the randomly placed agents shows that the optimization process using any of the three algorithms did improve fitness and reduce mortality over the random-placement approach.

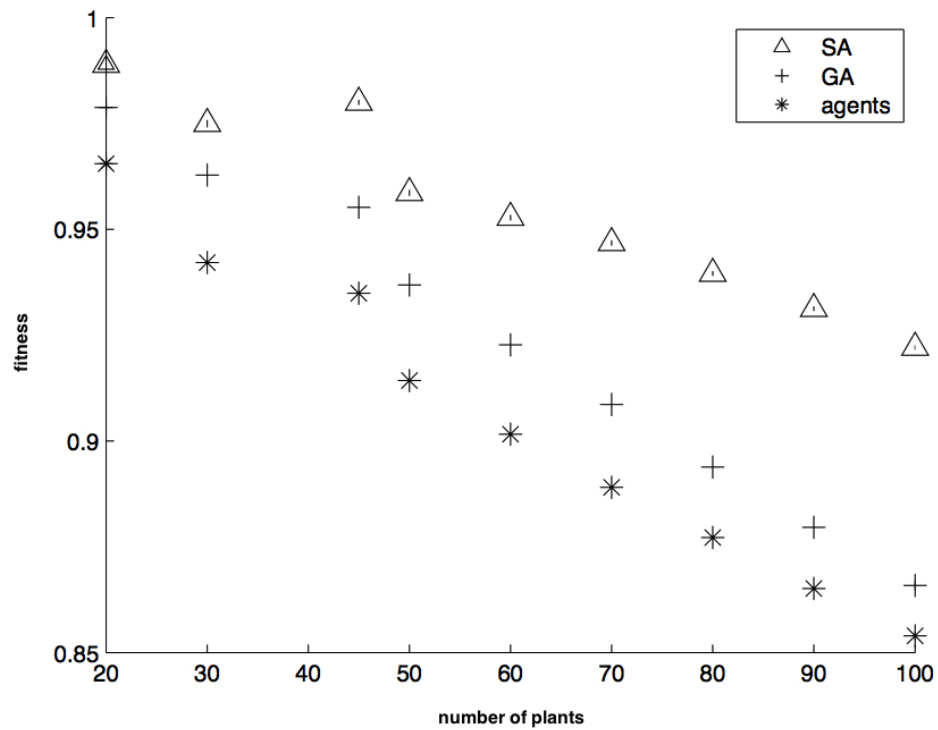


Figure 9.1: Mean fitness scores for each optimization strategy for 20 to 100 plant agents (half from each species) on the simulated landscape with the light and water properties shown in Figures 8.1 and 8.2. SA consistently produced the highest fitness of the three strategies. However, the difference in the fitness scores for these strategies was less than 10% on all plant collections.

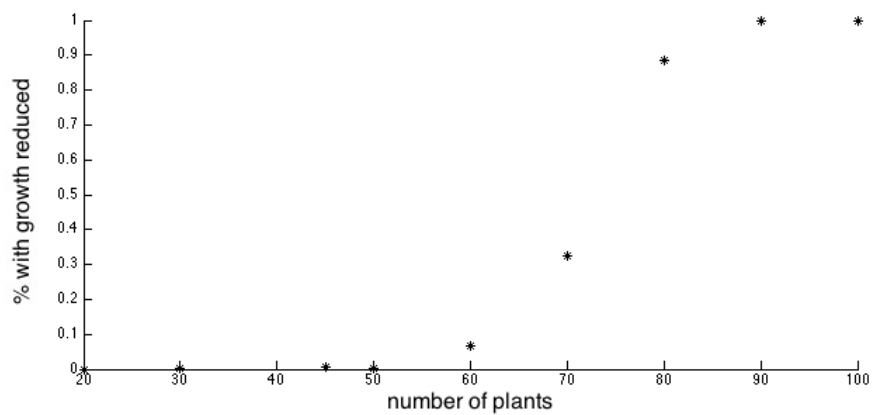


Figure 9.2: Percentage of the plant agents of the experiment in Figure 9.1 that experienced a reduction in growth due to competition for water with nearby agents. As more plants were added to the landscape, competition increased, lowering the average fitness of the plants. However, the increase was not linear. There was limited competition with fewer than 60 plants on the landscape, and the competition increased sharply between 60 and 80 plants. With 80 or more plants, all plants experienced a fitness reduction due to competition.

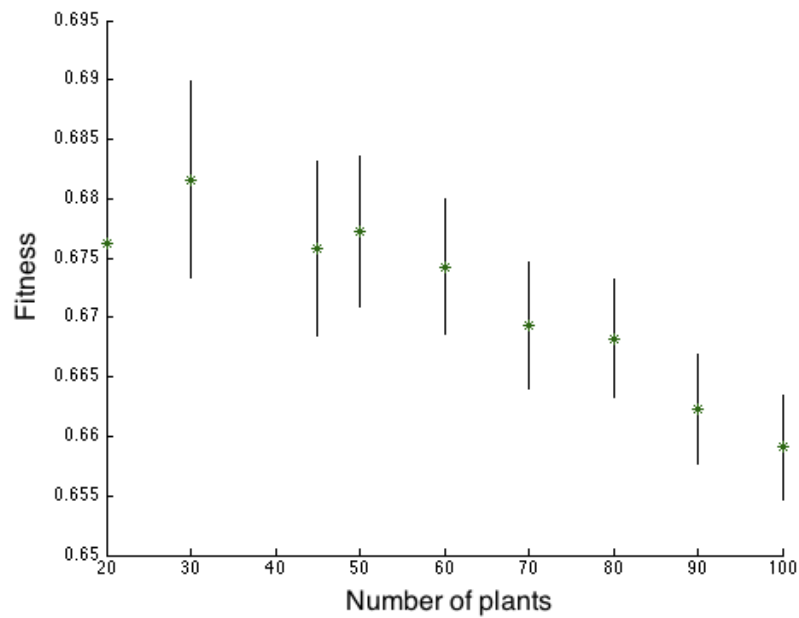


Figure 9.3: Mean fitness scores (with 95% confidence interval) for plant agents in the random arrangements on the landscape shown in Figures 8.1 and 8.2. In these simulations, a random location was selected for each plant agent and the fitness score was calculated at that location. Just as in the optimized arrangements, fitness dropped as more plants were added to the landscape. Scores were lower overall than they were for the optimized arrangements. “Mortality” (scores below the 0.45 survival threshold) on the landscape was also higher with random placements than for the optimized arrangements.

9.2 Fitness function α and β parameters and water application efficiency

The fitness function defined in this thesis uses two parameters α and β to weight the significance given to growth and water application efficiency (WAE): $f = \alpha \cdot growth + \beta \cdot WAE$. Prioritizing *growth* over *WAE* ($\alpha > \beta$) means that finding the light and water conditions that maximize an agent's growth has higher priority than finding the conditions that provide the most growth for the least amount of water. To evaluate the effects of this strategy on the characteristics of the optimized solutions, simulations explored how α and β affected the *WAE* and competition for water for the individual agents, as well as how these individual measures affected the total water applied to the landscape.

Increasing the value of the β parameter (and decreasing α) is designed to push agents into a location where the watering frequency at that location maximizes WAE for the individual agent. This behavior was verified in these simulations. Figure 9.4 shows the mean watering frequency as a function of β , for the 20-plant SA solutions. On the y-axis, 1200 mL is the mean daily watering amount for the high watering frequency, 600 mL is the mean daily watering amount for the medium frequency, 400 mL is mean daily watering amount for the low frequency, and 300 is the mean daily watering amount for the very low watering frequency. This graph shows that increasing β did indeed push the plant agents of both species from a watering frequency optimized for *growth* when β was low to one optimized for *WAE* when β was high. When β was low ($\beta < 0.5$), the mean watering frequency for the marigolds was just under 1200 mL: the daily watering conditions that produced the highest growth for this species in the greenhouse experiments and the highest *growth* score in the model. When $\beta > 0.5$, the observed watering frequency for the marigold agents dropped to 600 mL daily, the frequency where the maximum *WAE* score was observed. The same pattern existed for the impatiens agents. The mean watering frequency of 600 mL at the agents' locations observed when $\beta < 0.4$ was optimal for *growth* for this species. The watering frequency of 400 mL, which maximized *WAE*, was observed when $\beta > 0.4$.

As more plants were added to the landscape, the effect of β on the watering frequency of the

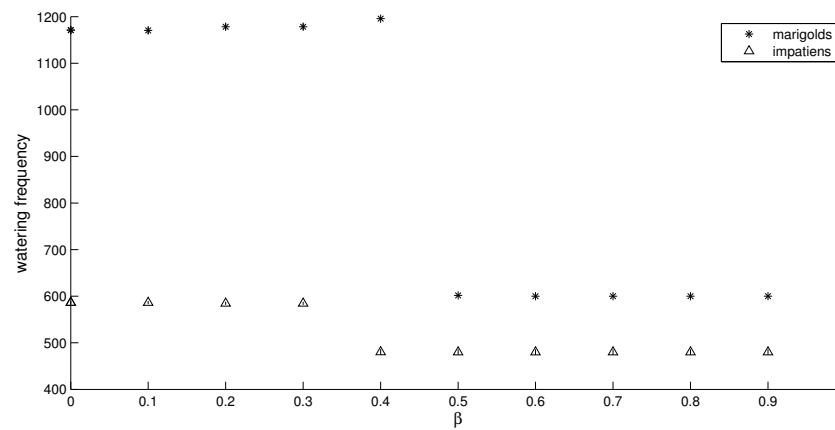
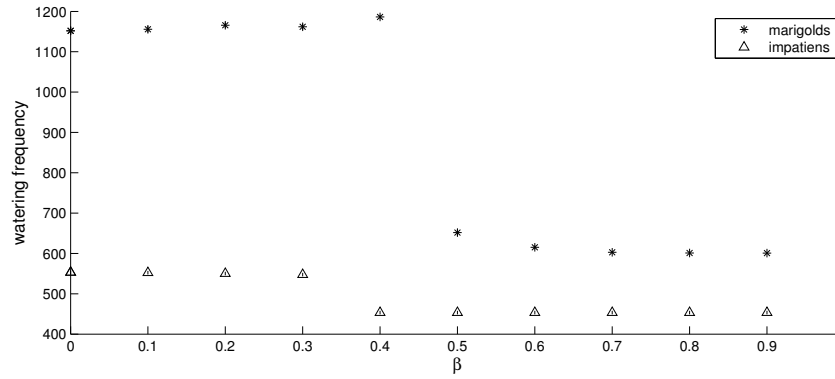


Figure 9.4: Mean watering frequency at the locations of all plant agents in the 20-plant SA solutions. This graph shows that increasing β decreased the required watering frequency for individual plant agents. There was a sharp drop in the watering frequency when $\beta \geq 0.5$, which places equal or greater priority on WAE over $growth$ in the fitness function: $fitness = \alpha \cdot growth + \beta \cdot WAE$.

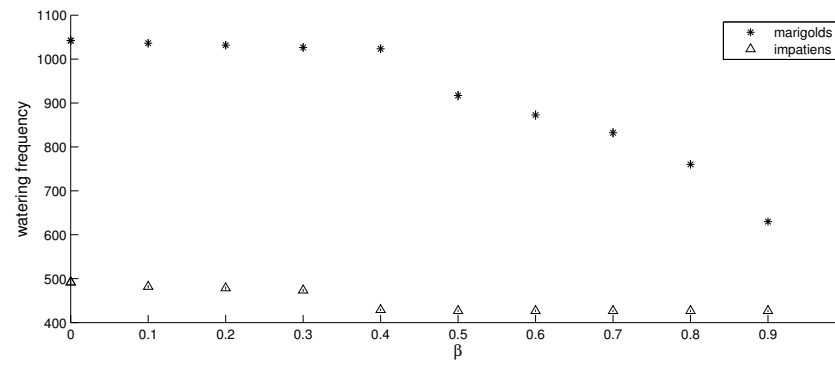
individual agents changed. Figure 9.5 shows the watering frequency for each species for collections of 30, 60, and 80 plant agents, as a function of β in the SA solutions. With 30 plant agents, the pattern was similar to that observed with 20 plants — there was a shift in the watering frequency from the one that maximized *growth* to the one that maximized *WAE*. With 60 plants on the landscape, the shift in watering frequency was less pronounced for the marigold agents. There were fewer plant agents of this species in the optimal watering frequency even when $\beta < 0.5$, as indicated by the lower mean watering frequencies for both species as compared to the 30-plant results. A value of $\beta \geq 0.9$ was required to shift the plants to the optimal *WAE* watering frequency.

The observed behavior with 60 plant agents and both high and low β values was due to the same thing — competition on the landscape. There was not enough space to provide optimal light and water conditions for 60 plant agents. In the optimal *WAE* watering frequency for the marigold agents, this species needed additional space to offset the growth reduction from the reduced watering. However, with 60 plant agents, this space was not available and a higher β value was needed to offset the reduced *growth* portion of the fitness score that plant agents experienced due to competition. This conclusion can be supported by examining the number of neighbor competitors that plant agents had as a function of β , as shown in Figure 9.6.

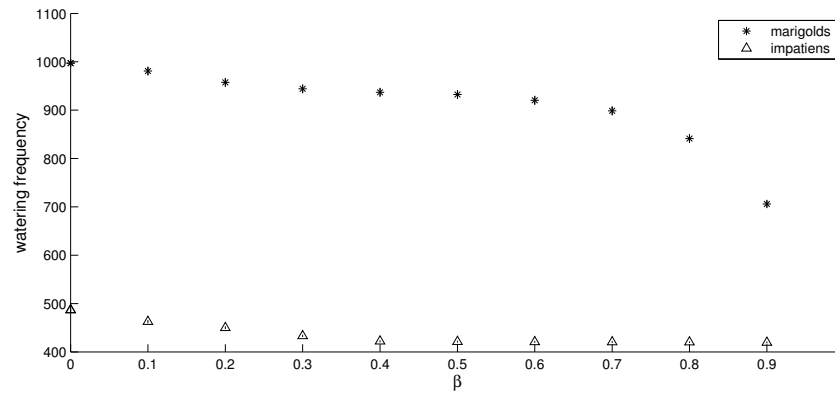
With 30 plants on the landscape, increasing β decreased the number of neighbor competitors. With low β , plant agents were confined to areas with a watering frequency where additional space was not needed and competition could not reduce growth. With high β , there was ample room for agents to spread out in the lower watering frequency region without having to compete for water with their neighbors. However, with 60 plants on the landscape, a different pattern emerged. The number of neighbor competitors dropped for β values between 0.6 and 0.8, indicating that agents were being located in multiple watering frequency zones, a conclusion that is also supported by the mean watering frequency for these β values. The number of competitors then increased sharply for $\beta \geq 0.8$ as the weight applied to *WAE* was high enough to offset the reduced fitness generated by competition for water in the lower watering frequency region. When $\beta = 1.0$, fitness was entirely determined by *WAE* and the number of competitors was not included in the fitness calculation.



(a) 30 Plants



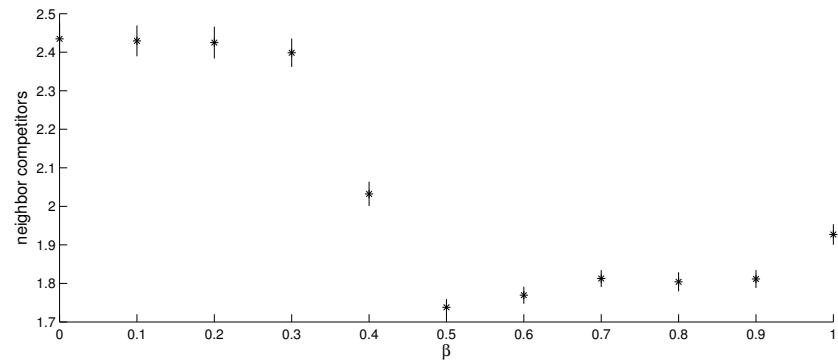
(b) 60 Plants



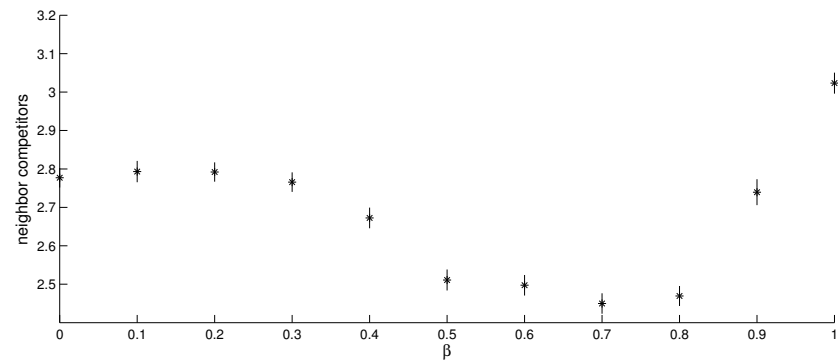
(c) 80 Plants

Figure 9.5: Mean watering frequency at the locations of all plants in the 30-, 60-, and 80-plant solutions for SA. This graph shows that increasing β decreased the required watering frequency for individual plants, and that the watering frequency and the effect of β also depended upon the number of plants on the landscape. With fewer plants, increasing β caused a sharp drop in the watering frequency. With 80 plants, watering frequency was stable until $\beta \geq 0.8$.

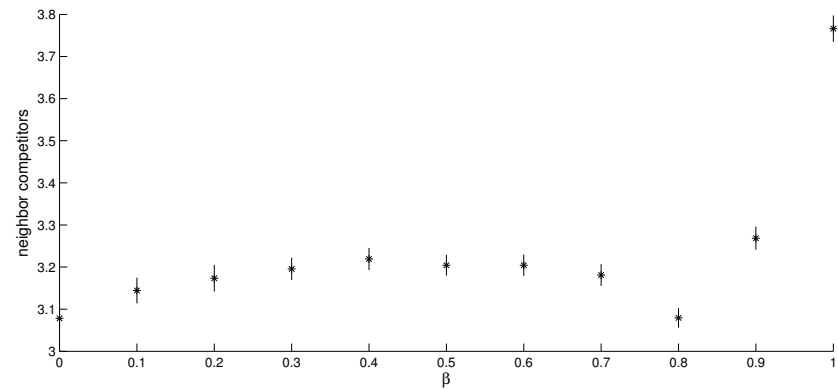
The effects of crowding increased in the 80-plant solutions. Neighbor competitors were greater than three (the threshold for a growth reduction established in Chapter 5) for all values of β , reflecting that the landscape had reached a saturation point and all agents experienced competition in the lower watering frequency zone. With 80 plants, shifting the watering frequency from *growth* optimization to *WAE* optimization required $\beta \geq 0.8$, which was similar to the behavior of the 60-plant solutions.



(a) 30 Plants



(b) 60 Plants



(c) 80 Plants

Figure 9.6: The mean number of neighbor competitors (with 95% confidence interval) when 30-, 60- and 80-plant agent SA solutions. With 30 plant agents, the number of neighbors decreased as the value of β increased: there was room on the landscape for all plants in the lower watering frequency regions. However, as more plant agents were added to the landscape, competition for water increased due to the higher number of neighbor competitors in the 60- and 80- plant agent solutions. A high β value was needed for WAE to offset the growth reduction from competition.

9.3 How agents' watering frequency and location affected the emergent water applied

The required watering frequency and location of the individual plant agents determined the total amount of water that needed to be applied to the landscape. The method for calculating water applied was shown in Equation (8.1), and can be summarized as: at each watering interval, water every cell with a plant agent and every cell around a plant agent, but do not water a cell more than once. The water applied to the landscape involves a trade-off — plants need space around them to grow in lower watering frequency conditions. However, with additional spacing, more cells need to be watered. In the simulations performed here, the amount of water applied involved a complex interaction that included the individual watering frequency, competition, and the varied influence of β for different plant populations. The mean water applied for 20-plant, agent-search solutions is shown in Figure 9.7. The x-axis is the value of β and the y-axis value is the total water applied to the landscape over 12 days. In these simulations, the reduced watering frequency of the individual agents when $\beta \geq 0.5$ translated into a reduction in the mean water applied of approximately 10%.

In the individual solutions produced using the same β value, the spacing between plant agents resulted in a range of values for the water emergent water applied. Figure 9.8 shows the water applied in all individual solutions for each optimization strategy for a representative selection of β values ($\beta \in [0.0, 0.3, 0.8, 1.0]$). For the SA solutions, water applied was as low as 2400 mL when $\beta = 1.0$ and as high as 4200 mL when $\beta = 0.0$, a difference of 43%. For the $\beta = 1.0$ results, solutions with similar fitness used between 2400 mL and 3350 mL, a difference of almost 30%. These results show how the positions of individual plant agents and different levels of competition with nearby plant agents, as discussed in previous sections, affect the emergent behavior on the landscape. Plant agents shared water by clustering together. Unless the level of competition was above a threshold that reduced fitness, all agents had the same fitness, but the water applied to the landscape was reduced.

The effect of β on the emergent water applied depended upon the number of plant agents on the landscape. Figure 9.9 shows the mean water applied for all plant agent collections for all

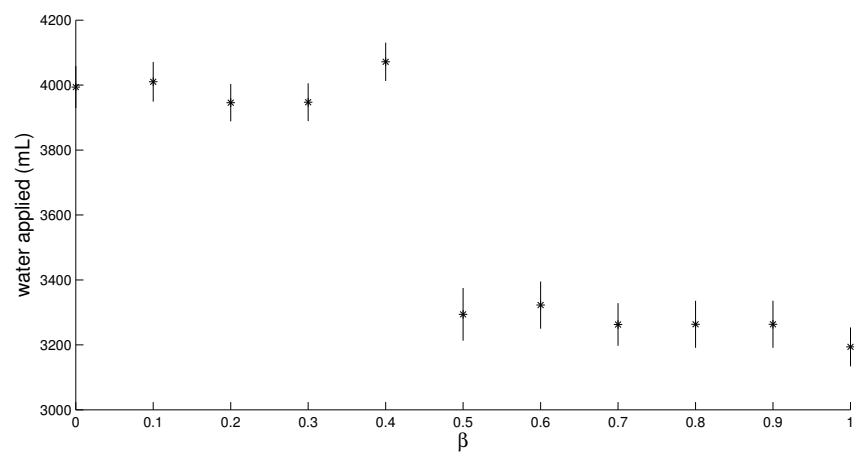
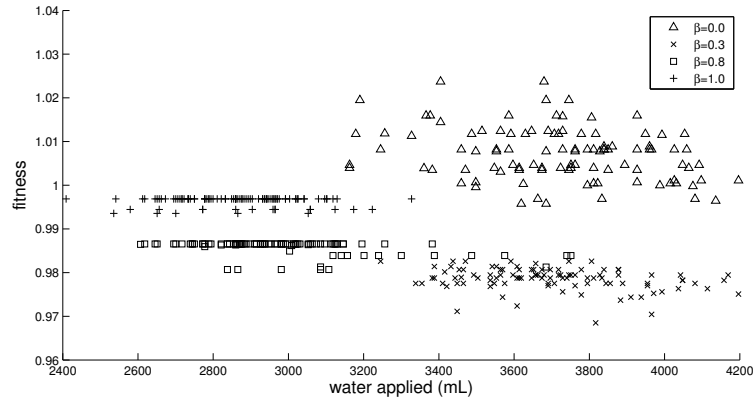
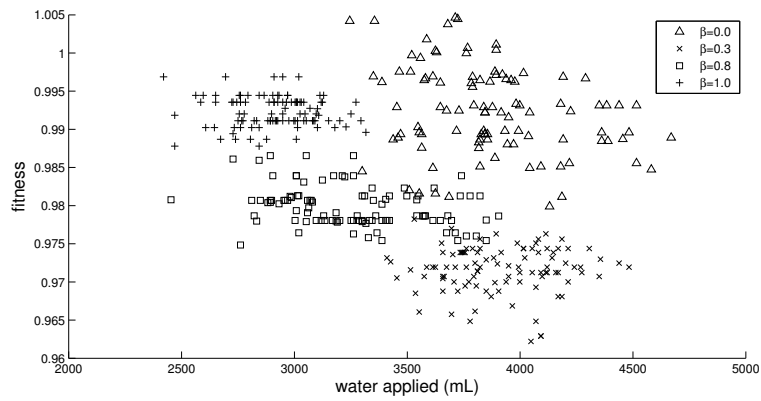


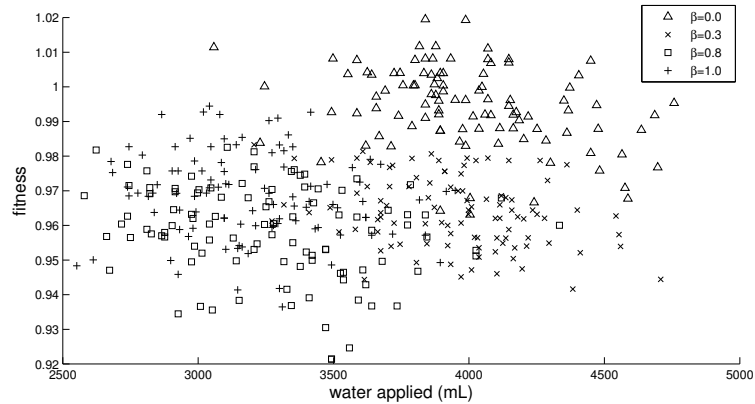
Figure 9.7: Mean water applied (with 95% confidence interval) to the landscape for all β values for the 20-plant collection arrangements produced using agent search. The water applied to the entire landscape follows the same pattern as the watering frequency of individual agents: increasing β reduces water use for both the individual plants and the entire landscape.



(a) SA



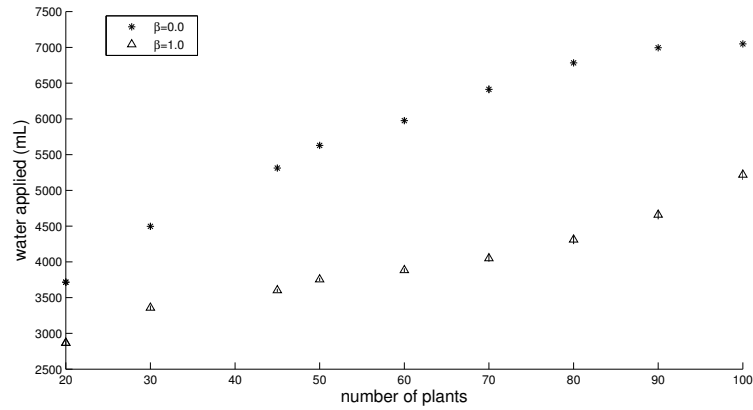
(b) GA



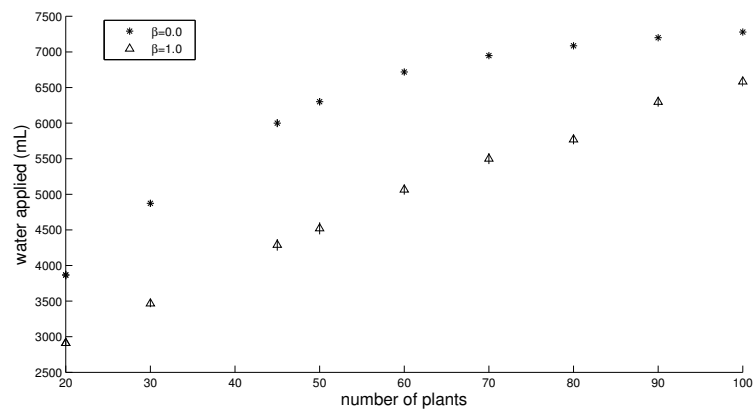
(c) Agents

Figure 9.8: Water applied vs. fitness for the 20-plant solutions for SA, GA, and agent search for representative β values. In the SA solutions, there was a clear delineation in the water applied for different values of β . For the GA and agent solutions, the pattern was more variable, but still showed a trend of reduced water and fitness with increasing β . The variability in the water applied for the same fitness score in solutions for all three optimization strategies was due to the effects of spacing on the number of cells to water and competition for that water.

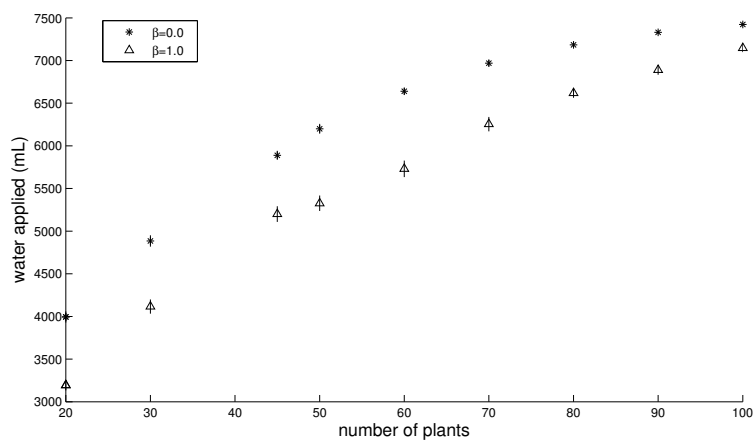
three optimization strategies. These results show that increasing β was most effective at reducing water applied when there were 50 to 70 plant agents on the landscape. This was true for all three strategies. The difference between $\beta = 0.0$ and $\beta = 1.0$ was strongest in the SA solutions, where there was also a marginal increase in water applied between 50 and 70 plant agents. In all three strategies, the difference between these two values of β decreased with 100 plant agents on the landscape. At this concentration, the landscape was crowded and most cells needed to be watered, showing the limits of the optimization for this landscape and these collections of plant agents.



(a) SA



(b) GA



(c) Agents

Figure 9.9: Mean water applied (with 95% confidence interval) for all three optimization strategies. These plots show the difference in the water needed by the different solutions when the priority is on *growth* ($\beta = 0.0$) or *WAE* ($\beta = 1.0$). In the SA solutions, the curves for the two values of β diverge around 50 to 70 plant agents, where the optimization was most effective at reducing water use on the landscape. For the GA and agent solutions, prioritizing *WAE* also reduced water applied, but not as significantly as in the SA solutions.

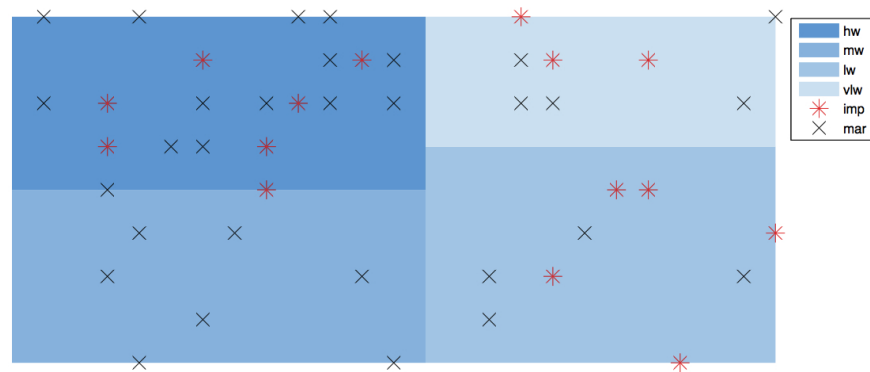
9.4 Greenhouse validation experiment

To validate the model presented in this thesis, two arrangements were selected for planting in the greenhouse using live plants. This experiment involved a random arrangement and one generated by agent search, both shown in Figure 9.10, with the priority on *WAE* in the fitness function. Plants were watered daily in the amounts predicted by the model for water applied for 17 days¹. Unfortunately, there was an error in planting and arrangements were flipped horizontally from the optimized arrangements, i.e. plants in the high watering frequency region should be in the very low water zone, plants in the medium-water zone should be in the low-water zone, and vice versa. The planted arrangements are shown in Figure 9.10. Even with this mix up, there were interesting results from this experiment.

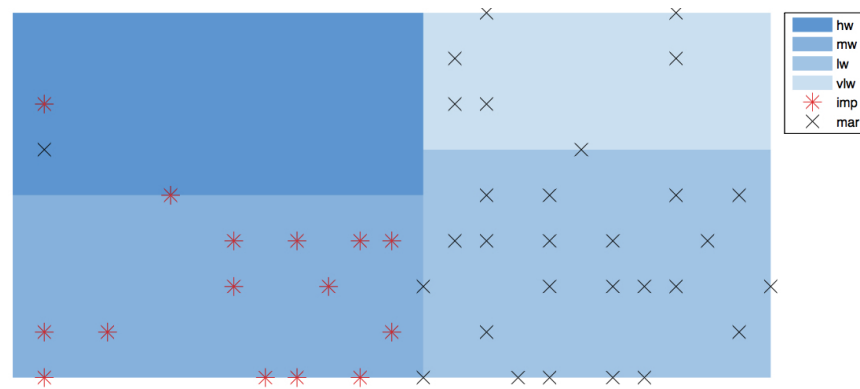
The arrangement that was intended to reflect the optimized solution, although not technically correct due to the planting error, will be referred to here as the optimized arrangement for convenience. This arrangement required 5,560 mL of water. The random arrangement required 8,580 mL of water according to the model, and this was the amount applied to the physical plants in both trays. The 35% difference in water applied between the optimized and random arrangements resulted in an equivalent reduction in biomass in the optimized tray — 0.086 grams as compared to 0.132 grams in the random tray. However, there was zero mortality in both arrangements, and all plants appeared healthy throughout the experiment, as shown in Figure 9.11.

In the optimized and random arrangements, the additional water applied to the random arrangement generated a greater growth increase for the marigolds than for the impatiens. The marigolds had a mean biomass of 0.106 and 0.171 grams respectively, a difference of 38%, and the impatiens had biomass values of 0.047 grams in the optimized and 0.053 grams in the random arrangements, a difference of 12%. This result is in line with the model predictions. The marigolds are a high-water species that benefitted from daily watering, while the impatiens performed better

¹ The experiment was originally intended to run for three weeks, which was the length of time for the other greenhouse experiments. However, when it became clear that plants were starting to compete, the experiment was stopped.



(a) Random arrangement



(b) Optimized arrangement

Figure 9.10: Random and optimized arrangements for the validation experiment with live plants. The optimized arrangement was selected to conserve water, but to still achieve zero mortality throughout the experiment.

when they were watered every other day. It is worth noting that, for both species, these biomass values were greater than those observed in the experiments described in Chapter 5, which were used to generate the growth curves for both species. This is most likely due to the difference in the time of year when the experiments were performed. The growth curves were generated in the winter and this validation experiment was performed in the spring with longer photoperiod and warmer temperatures.

Another interesting result from this validation experiment was how the fitness scores for each species correlated to the growth for that species. For the marigolds, the difference in fitness — 0.75 for the random arrangement and 0.56 for the optimized arrangement — reflected the difference in growth due to the effect of the mis-planting. The marigolds in the optimized arrangement, as planted, were primarily in the low-water conditions. However, they should have been in medium-water conditions, which would have used more water but also provided higher fitness scores and growth. The low fitness score, therefore, actually supports the model. Marigolds in the low watering frequency should have low growth and correspondingly low fitness scores. For the impatiens, however, the fitness score did not reflect the difference in growth. The mean fitness score of these plants in the random arrangement was 0.66 and in the optimized arrangement was 0.75. In the random arrangement, this species was placed in the high, medium, and low watering frequencies. In the model, impatiens in the high watering frequency experienced a substantial growth reduction compared to impatiens in the low and medium watering frequencies. In this validation experiment, impatiens in the high-water zone did not experience the same growth reduction. This result could be due to the time of year. The amount of water applied in the high watering frequency region did not saturate the soil in the same way that it did in the winter when the model was developed. As a result, plants in this watering frequency did not experience reduced growth, but the fitness score was still low.



(a) Random arrangement



(b) Optimized arrangement

Figure 9.11: Random and optimized arrangements on the final day of the validation experiment with live plants. The optimized arrangement used 35% less water than the random arrangement but also produced equivalently less biomass.

9.5 The effects of optimization-strategy parameters on fitness scores and runtime

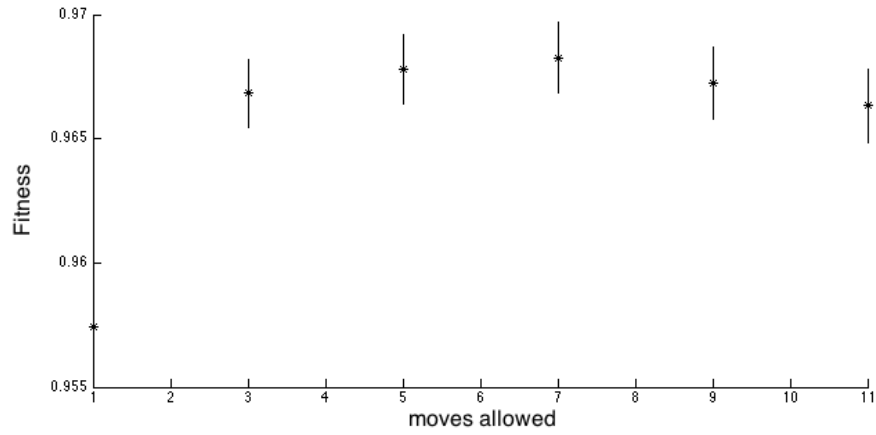
The evaluation of how each optimization strategy performed that was presented in the previous sections used the parameter combination that produced the highest fitness scores for that strategy. However, the parameter values affected the runtime of the algorithm. In this section, these effects are evaluated.

9.5.1 Agent search

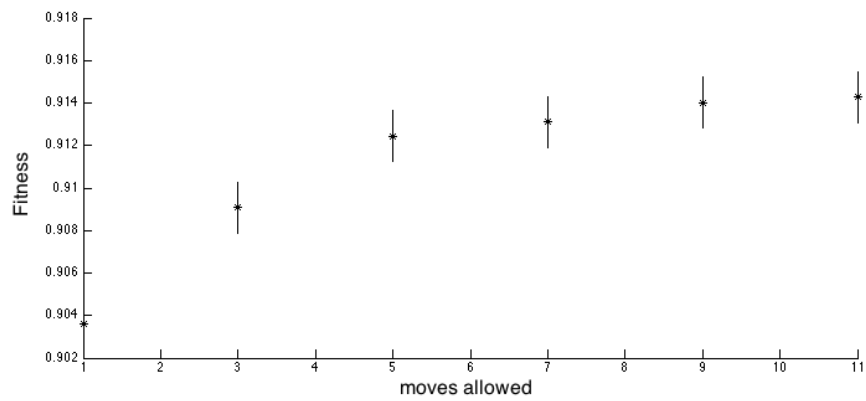
The parameters in the agent-based search routine evaluated here were the local search radius around each agent and the number of times each agent could move to improve its fitness score. This search routine was evaluated using 1 to 15 moves allowed and a local search radius of 0 to 3 cells. Figure 9.12 shows how the number of moves allowed affected fitness in the 20-, 50-, and 80-plant solutions. Each data point in this figure involved a local search radius of three cells. The algorithm converged after three moves allowed in the 20-plant solutions, five moves allowed in the 50-plant solutions, and seven moves allowed in the 80-plant solutions. Increasing the number of plant agents also increased the moves allowed needed for convergence.

The other parameter modified in the agent-search routine was the local search radius, the number of cells in each direction the agent examined for a better location before jumping to a random location on the landscape. A search radius of zero meant that agents could only make random jumps. This radius produced the lowest fitness scores — only slightly higher than those produced through random placements regardless of the number of moves allowed. For example, the 20-plant solutions had a mean fitness score of 0.75 with a zero-cell search radius and a mean fitness of 0.96 with a one-cell search radius. A similar pattern was observed for the local search radius — allowing a one-cell local search radius improved fitness, but a larger radius had little effect on fitness scores.

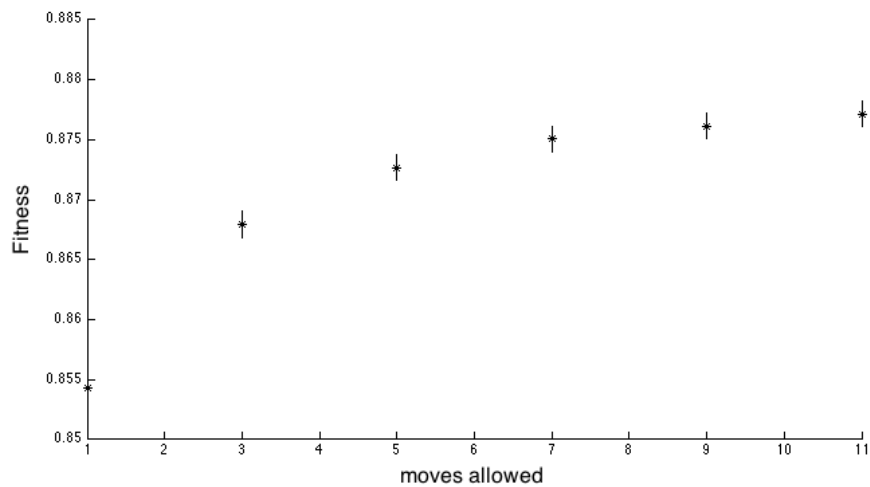
In summary, the agent search converged to a solution using only a few moves allowed and a small value for the local search radius. Additional moves allowed and a larger search radius did not



(a) 20 Plants



(b) 50 Plants



(c) 80 Plants

Figure 9.12: Mean fitness (with 95% confidence interval) vs. moves allowed in the 20-, 50-, and 80-plant agent solutions produced by the agent search algorithm. In the solutions shown here, allowing agents to make even one move significantly improved fitness over the random placements. However, additional moves allowed did not produce additional fitness improvements. As the number of plant agents on the landscape increased, so did the number of moves allowed that were needed for the algorithm to converge to a solution.

improve solution fitness scores. This behavior has some advantages and some disadvantages. The algorithm ran faster than the other two strategies and produced fitness scores higher than those of random placements. However, the disadvantage was that the fitness scores were lower than those produced by the other two algorithms.

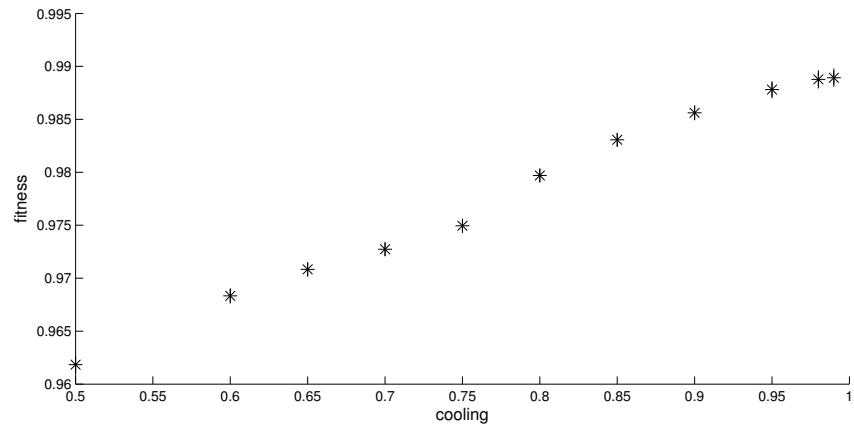
9.5.2 Simulated annealing

For simulated annealing (SA), the operative parameter was the cooling pressure — the α value in the $T = \alpha T$ equation that controlled the likelihood of accepting solutions that did not improve global fitness for all plant agents on the landscape. Values for α between 0.50 and 0.99 were evaluated: the effects of those values on fitness is shown in Figure 9.13 for the 20-, 50-, and 80-plant solutions. Fitness scores were correlated with the value of α , i.e. increasing α allowed the algorithm to run longer and explore more of the search space, producing higher fitness scores. In all solutions, there was an upper limit on fitness scores: increasing α above 0.95 did not substantially increase fitness scores. This behavior was different than agent search behavior where more moves allowed and a larger search radius did not necessarily increase fitness scores.

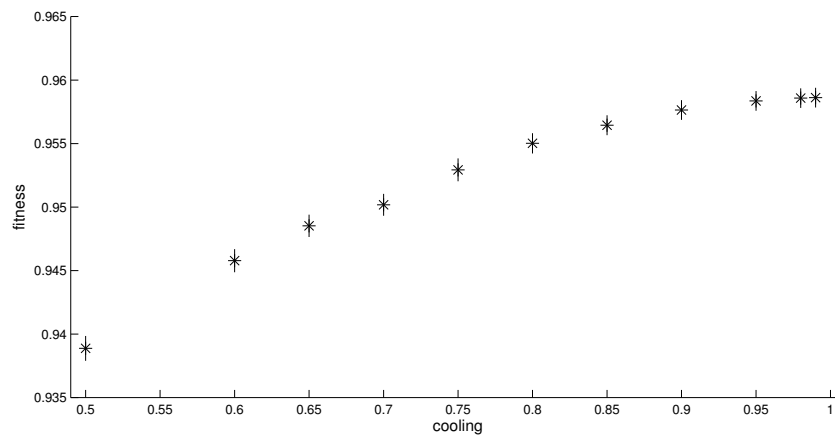
9.5.3 Genetic algorithm

For the genetic algorithm (GA), the effects of the number of iterations and the number of mutations on each iteration were evaluated. The former was varied between 300 and 1700 and the latter was varied between zero and seven. For all plant-agent collections, fitness increased with increasing number of iterations, similar to the SA behavior. The fitness scores for the 20-, 50-, and 80-plant agent solutions using one mutation are shown in Figure 9.14. At approximately 1300 iterations, the increase in fitness leveled off. Additional iterations resulted in modest increases in fitness scores.

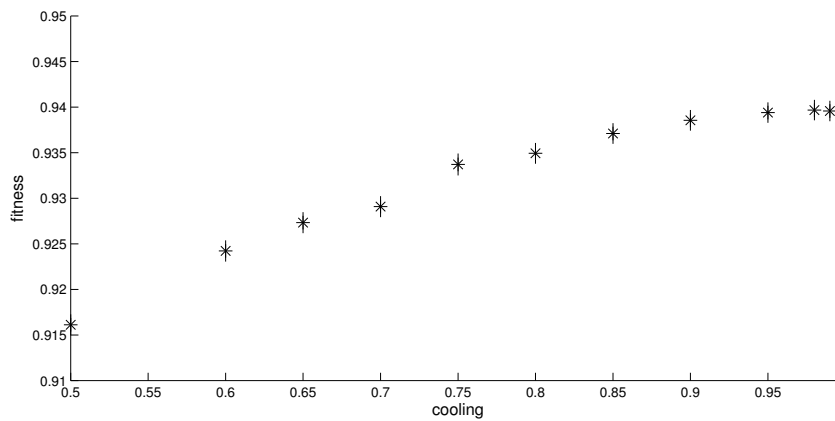
The other parameter evaluated was the number of mutations. In these solutions, one mutation produced higher fitness than zero mutations, but beyond that, more mutations decreased fitness. For example, in the 45-plant solutions, fitness dropped from 0.96 with one mutation to 0.91 with seven



(a) 20 Plants



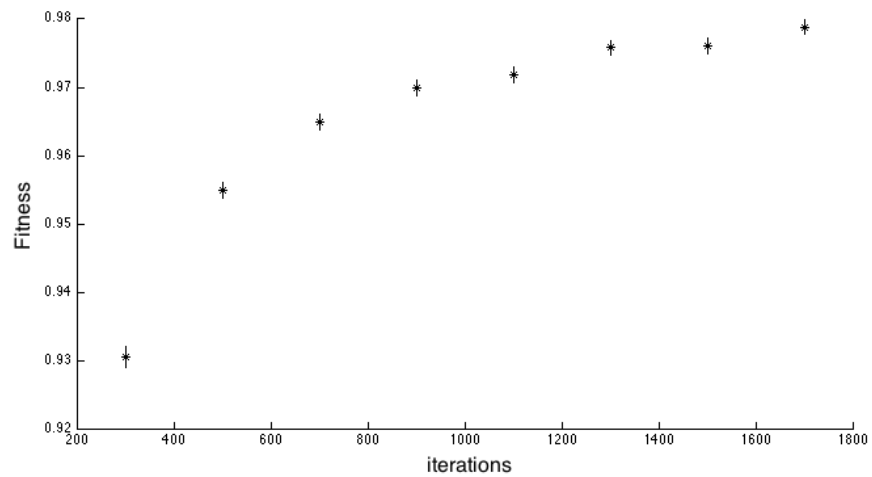
(b) 50 Plants



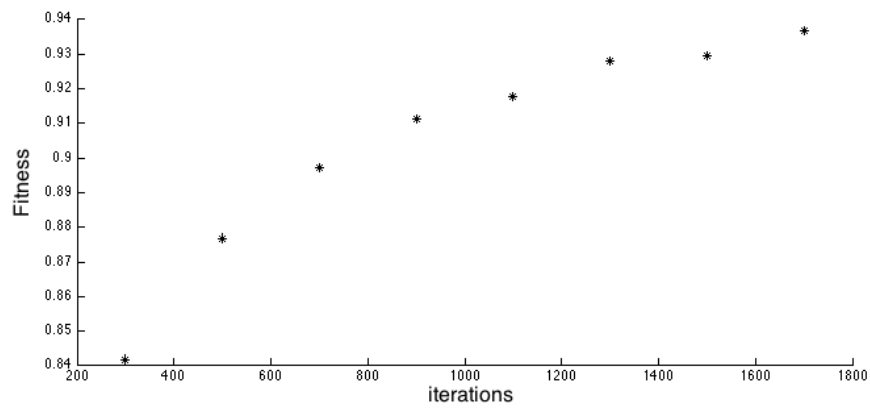
(c) 80 Plants

Figure 9.13: Mean fitness scores (with 95% confidence interval) vs. cooling value for 20-, 50-, and 80-plant solutions for SA. Allowing the search algorithm to run longer — by using a higher value for the cooling parameter — produced higher mean fitness scores for each of the plant collections.

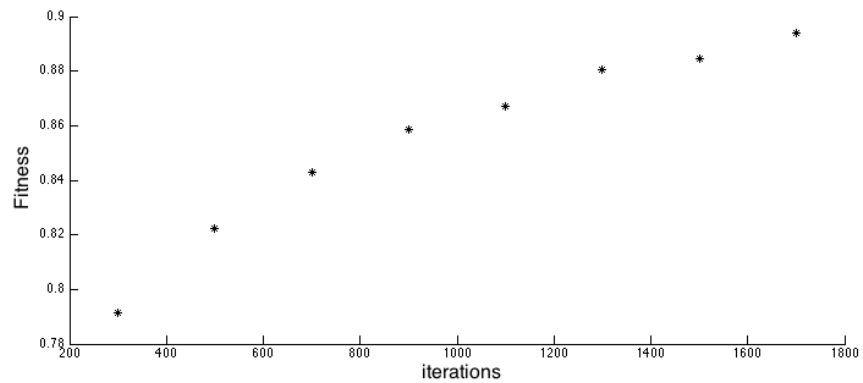
mutations. Recall that these mutations are designed to introduce randomness into the algorithm and prevent the search from getting stuck in local optima. In this model, there is another feature that has a similar effect: the repair method used when two plant agents are placed in the same location during crossover. Step 9 of Algorithm 3 selects a new random location for one of the plant agents, effectively adding additional mutation to the population.



(a) 20 Plants



(b) 50 Plants



(c) 80 Plants

Figure 9.14: Mean fitness scores (with 95% confidence interval) for the 20-, 50-, and 80-plant collections for 300 to 1800 iterations optimized by the GA. The number of iterations determined how long the algorithm searched for a solution. Allowing the algorithm to run longer improved solution fitness, but this effect slowed at around 1300 iterations.

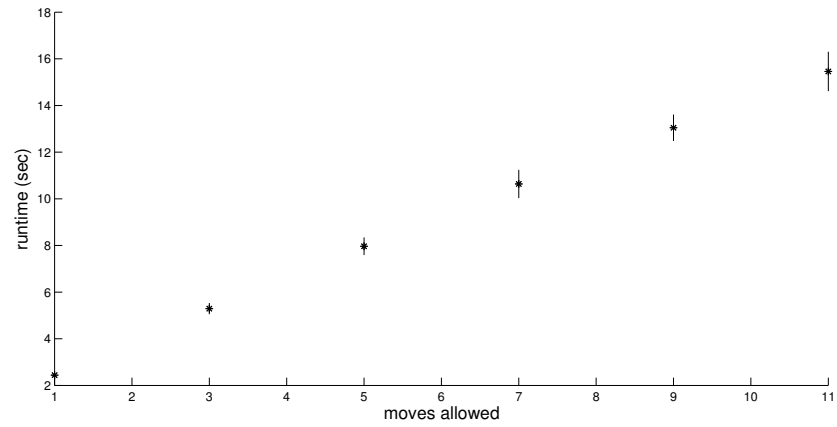
9.6 Runtime comparison of the optimization strategies

For certain parameter combinations, each of the optimization strategies produced, on the average, similar fitness scores. For example, the GA with one mutation and 1100 iterations produced the same fitness scores as agent search with five moves and a three-cell local search radius. The parameter combinations also affected runtime, defined as the user CPU time in seconds on the Janus supercomputer required to execute the algorithm. It is important to consider runtime as part of the evaluation. The agent search, for instance, outperformed the GA in the sense that it produced comparable fitness scores in less time.

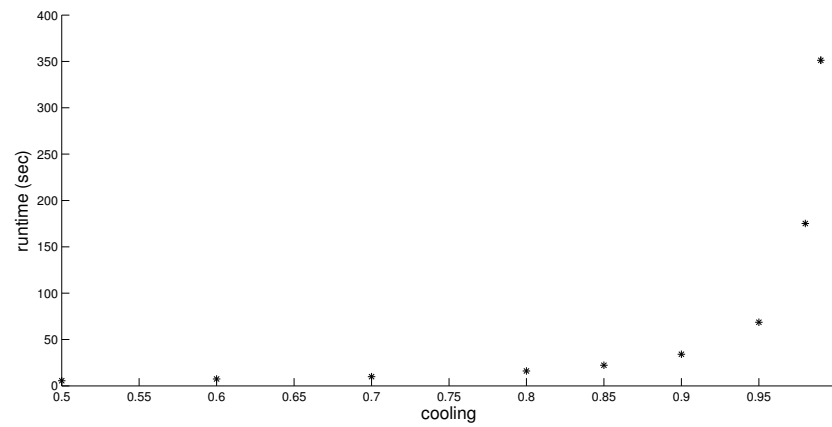
The runtime in seconds for each of the three algorithms for the 20-plant solutions is shown in Figure 9.15. In this case, agent search used a local search radius of three cells and the GA used one mutation. For both agent search and the GA, the runtime increased linearly with the number of moves and iterations. For all parameter combinations, agent search was an order of magnitude faster than the GA. The minimum runtime for each strategy was two seconds for agent search and 25 seconds for the GA. These values were generated using one move and a three-cell local search radius for agent search, and 300 iterations and one mutation for the GA. The fitness scores associated with these parameters were 0.95 for agent search and 0.93 for the GA. Allowing two additional moves for agent search improved fitness to 0.965 and only increased runtime to 6 seconds, still less than 25% of the time required for the GA. This same pattern was observed on larger problems for these two algorithms. On the 50-plant solutions, agent search required a maximum of 80 seconds to produce solutions with an average fitness of 0.91. In comparable runtime, the GA in a comparable runtime produced solutions with an average fitness of 0.84.

For SA, the runtime did not increase linearly, but rather remained constant up to a cooling (α) value of 0.85, then increased sharply above that value. The fitness improvement between cooling values of 0.90 and 0.99 was modest in all solutions, as shown in Figure 9.13. Although a cooling value of 0.99 produced the highest fitness scores, a cooling value of 0.85 produced fitness scores slightly higher than agent search in a comparable runtime. For the 20-plant simulations, agent

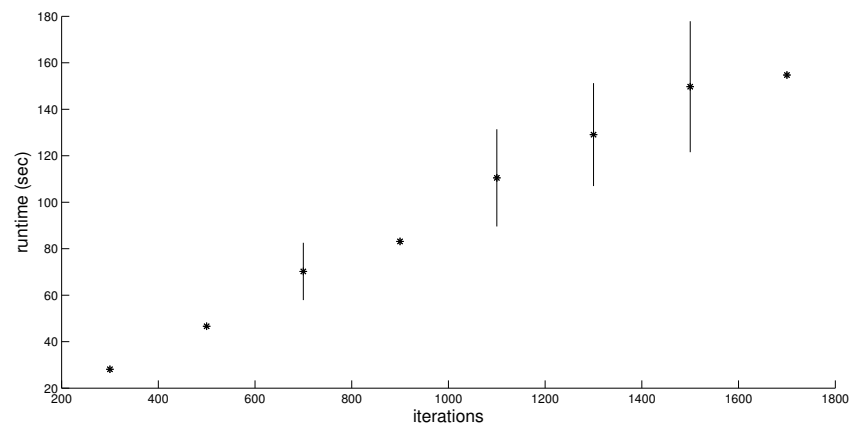
search produced a mean fitness score of 0.965 in 10 seconds and SA produced a mean fitness score of 0.98 in the same runtime. For the 50-plant simulations, the maximum mean fitness score for agent search was 0.914 produced in a runtime of 65 seconds. In a comparable time, SA produced a mean fitness score of 0.95. Therefore, in all solutions, SA produced a higher mean fitness scores than agent search in comparable runtime.



(a) Agents

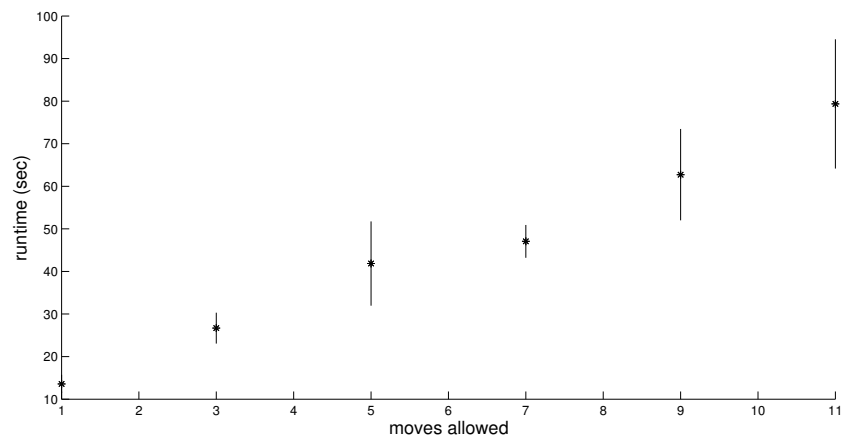


(b) SA

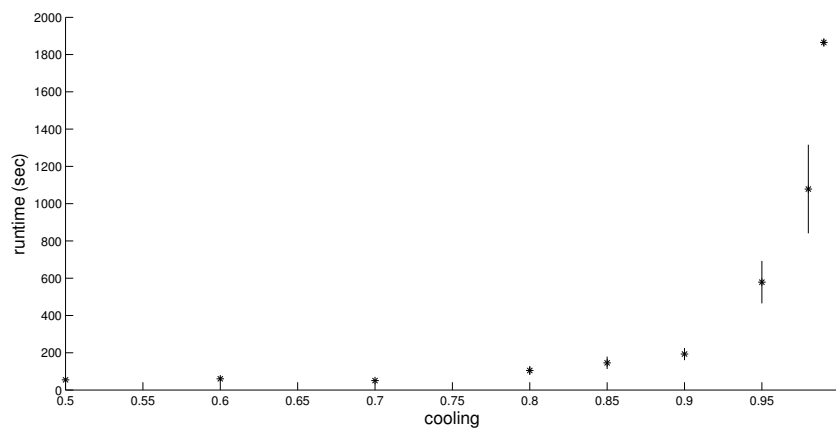


(c) GA

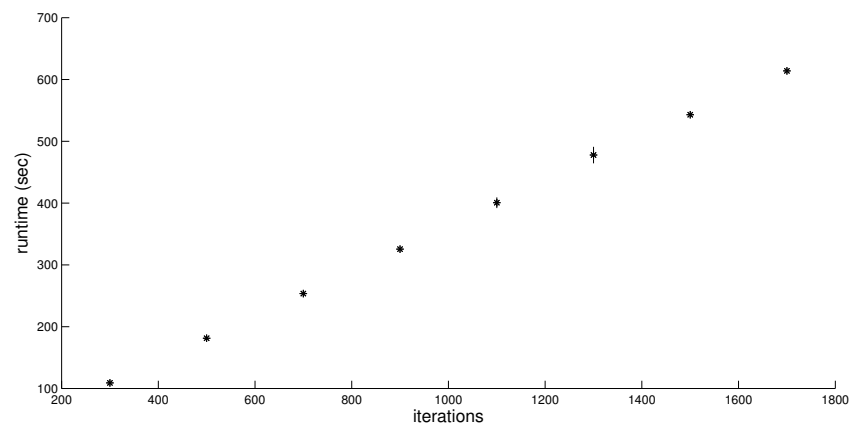
Figure 9.15: Mean runtime in seconds (with 95% confidence interval) for agent search, SA, and the GA for 20-plant solutions. Runtime was defined as the user CPU time to execute the algorithm on the Janus supercomputer. The agent search ran significantly faster than the other two algorithms while producing solutions with similar fitness scores.



(a) Agents



(b) SA



(c) GA

Figure 9.16: Mean runtime in seconds (with 95% confidence interval) for all three optimization strategies for 50 plants. Again, agent search ran faster than the other two algorithms in these tests.

9.7 Results from a commercial optimization package

The GA, SA, and agent search routines used in this thesis were implemented in C by the author of this work. Commercial software packages include GA and SA optimization routines that can be applied to solve the location problem addressed here. One example of such a package is the Matlab Global Optimization Toolbox. To use this toolbox, the user writes the fitness function that is called in the optimization process, and the Matlab routines offer several tunable parameters that the user can modify to affect how the optimization proceeds. In this section, the strengths and weaknesses of the author's handcrafted versions are compared to the Matlab commercial optimization package.

9.7.1 Setting up the Matlab GA

In the Matlab GA, the user defines the encoding — what each gene on the chromosome represents — and the fitness function. The objective in all of the optimizations presented in this thesis, thus far, was to *maximize* plant-agent fitness:

$$fitness_i = \alpha \cdot growth + \beta \cdot WAE \quad (9.1)$$

The only option in the Matlab GA is to *minimize* the fitness function. To apply this algorithm to the problem of optimizing plant placements, the fitness calculation for the individual plant agents was changed to be:

$$fitness_i = \frac{1}{\alpha \cdot growth + \beta \cdot WAE} \quad (9.2)$$

The encoding used here was the same as encoding used for the author's C implementation: each gene on the chromosome represented a plant, and the value of the gene was the plant's row and column position on the landscape. The Matlab GA only allows for 1D chromosomes, however, so the 2D landscape positions were converted to a 1D representation. Each gene had an integer value between 1 and $rows * cols$, where $rows$ was the number of rows on the landscape and $cols$ was the number of columns on the landscape. This value was then converted to the 2D coordinates

in the fitness function to calculate how the individual plants affected their surroundings. This encoding required a repair function similar to the one in the C implementation. The values for each gene were not unique, meaning that more than one plant could be placed at the same location. To correct this, the fitness function checked for plant agents in the same location and moved the lowest-scoring one to a new random location.

In the initial exploration of the Matlab GA, the default settings for most parameters were used. The exception was the mutation function, where the default settings produced solutions outside the feasibility range. This GA shares many features of the C implementation, but also includes some different ones:

- **Population size** — 20 chromosomes.
- **Crossover function** — Randomly select genes from each individual, which may or may not be contiguous, for the next generation.
- **Crossover fraction** — Replace 80% of the population in the next generation through crossover.
- **Generations** — Up to 100.
- **Mutation function** — Select 1% of the population and generate a new random location within the feasibility range (between 1 and *rows * cols*).
- **Parent selection** — Weight parents by their fitness and select high-performing parents at a higher rate than low-performing parents.

9.7.2 Why using the Matlab GA is better than implementing your own GA

There are many reasons why using a commercial optimization package such as the Matlab Global Optimization Toolbox is better than writing your own optimization routines. The Matlab package offers many GA features that would be time consuming to write from scratch and would

most likely require extensive debugging. For example, the C implementation included only the basic GA features, and for each of these features, only one option was available. There was only one crossover strategy, mutation strategy, population size, and parent-selection algorithm, for instance. In the Matlab GA, there are many options for each of these features, and switching between them is just a matter of changing an input parameter.

The fitness scores that the Matlab GA produced using the default parameter values also show the strength of a commercial package. These scores are shown in Figure 9.17 for the same plant collections and landscape used in previous solutions. In this figure, lower values for fitness are better than higher values since the GA works by minimizing the fitness function. The fitness scores in Figure 9.17 show the same pattern as those previously reported — fitness decreased as more plants were added to the landscape. While the fitness scores were not as good as those reported for the author's GA implementation, those reported scores were for the best parameter combination, discovered through extensive evaluation on the Janus supercomputer using code that took over six months to write. The time required to implement the fitness function in Matlab and run the GA was far smaller than the time required to implement the GA from scratch in C.

9.7.2.1 A different GA encoding

One potentially useful modification that would be simple to implement in Matlab and difficult to implement from scratch is a different encoding for the GA. In the encoding presented here, where each gene is a plant, the population size and composition are fixed. This allows the algorithm to optimize locations for a pre-defined population. Allowing the population to change presents another interesting optimization problem where the objective is to determine how many plants and *of what types* a landscape could support. The GA encoding could be easily changed in Matlab to let each gene represent a position on the landscape, and the value for the gene could be an integer representing the plant type.

These changes were implemented here as a demonstration of how this encoding might work. There were three values for each gene: 0 to represent no plant, 1 to represent a marigold, and 2

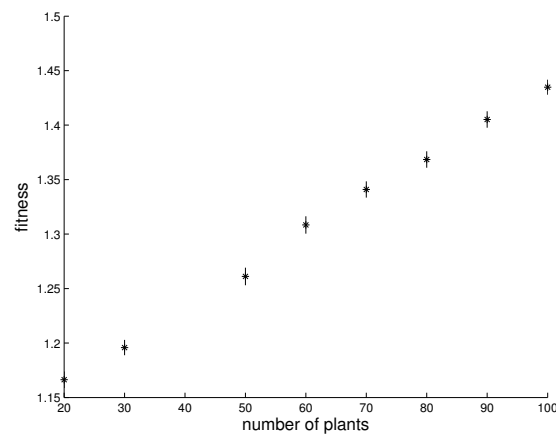


Figure 9.17: Mean fitness scores (with 95% confidence interval) for all plant-agent collections optimized using the Matlab GA with the default parameter settings. This results from this GA followed the same general pattern as the results from the author's GA implementation — fitness deteriorated as more plants were added to the landscape.

to represent an impatiens. The chromosome included 216 genes, representing the positions on the landscape. All other parameters were set to the defaults described previously. The average results from 100 solutions are shown in Figure 9.18. In these solutions, fitness was calculated as $\sum_i^n f_i$, meaning that adding more plants to the landscape potentially improved fitness scores. However, this figure shows that in the majority of the solutions, there were between 30 and 60 plants on the landscape, and as more plants were added to the landscape, fitness declined. There were only a few solutions with more than 70 plants and these solutions had poor fitness scores. This pattern was consistent with the results from the author's GA implementation where the optimization was most effective with 50 to 70 plants on the landscape.

This new encoding presents interesting options for exploring how to maximize the number and types of plants in a small space, which is discussed further in the Future Work section.

9.7.3 Limitations of the Matlab GA

There is a danger in selecting any commercial optimization package in that it is easy to believe that the software will just work and require little understanding from the user. Believing that one can just throw a software package such as Matlab at a problem neglects the complexity of understanding the features of the Matlab software or the role that the user needs to play in developing the fitness function. These pitfalls played out in some of the difficulties with applying the Matlab GA to this problem. There are many parameters that control the features of the GA. However, just having these features available does not guarantee good results if the user does not have an understanding of how the features work.

Another limitation of Matlab that is important in this problem is speed: Matlab is slow. On the 20-plant simulations, the Matlab GA required over 1660 seconds (11 minutes) to execute, almost five times longer than the longest-running optimizations using the C implementation. This limits the number of optimizations that can be performed. An exhaustive search of all possible input parameters for the GA would be difficult without access to a supercomputer. The runtime performance of the Matlab code can be improved by generating compiled .mex files. However, this

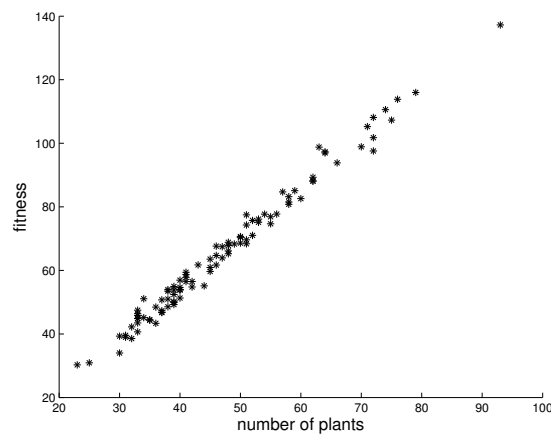


Figure 9.18: Fitness scores vs. number of plants using a GA encoding where the number of plants and plant types were not fixed. For most solutions, the optimized landscape included 30 and 60 plants. Fitness declined as more plants were added to the landscape, which was also observed in simulations where the number and type of plants were fixed.

approach requires additional programming skill: the .mex files are generated from user-supplied C code. A user would first have to write the fitness function in C and then convert it in Matlab. The complexity of this work potentially defeats the purpose of using the commercial package.

9.8 Future work

This thesis presented a new way of optimizing plant interactions for the purpose of water conservation on residential landscapes whereby the layout of the landscape is treated as a location problem. An agent-based model captured how plants performed in a set of light and water resources to assign a value to a location in a fitness function that measured how well the resources matched the plant agent's light and water requirements. These values were then used in different optimization strategies to maximize the fitness function output. An agent-based search routine developed as part of this thesis produced similar fitness scores to these established metaheuristics, but required less execution time.

Each of the elements in the framework presented here — the agent-based model, fitness function, and optimization strategy — presents options for future work. The framework is modular: an ABM can be easily swapped out for a different model while still using the same optimization strategy. Different optimization strategies can be applied using the same agent-based model.

Agent-based model In the model presented here, plant agents had two requirements for growth — light and water. Future work could include designing agents with other growth requirements or different growth variables. For example, plant agents with low-water requirements could easily be added by adjusting the water-response curve in the plant growth model. The same approach could be used to generate plant agents with different light requirements than those evaluated in this thesis. Other important growth variables could include nutrients. Adding these would involve adding additional parameters to the landscape to capture the nutrient levels at a location and then include the plant's response in the fitness function.

In this model, agents influence the surrounding light and water resources. Future work in this area could include changing how agents influence these resources. Species could generate

different degrees of shade by blocking more or less light than other species. This same approach could be applied to how agents compete for water and nutrients. Agents could have different competitive levels giving them preference over other nearby agents. All of this would make the model more realistic, which would have the benefit of being able to model more-complex systems and interactions.

One of the important and unusual contributions of this thesis is that the agent-based model was validated in the greenhouse. The additions to the model listed in the previous paragraphs could also be validated in future work. Studies could be performed outdoors using sensors to monitor the resources and determine how individual plants are influencing their surroundings. These findings could then be incorporated into a model for any species being studied.

Another intriguing approach to validation is crowd sourcing, which would involve the general public collecting data about plants growing on existing landscapes, including the conditions for those plants and how well they are growing. With enough data, growth curves could be generated for multiple species that capture how light, water, nutrients, and interactions with surrounding plants affect the growth of the species, as well as how the species affects its surroundings. Another avenue for this type of data collection might be to evaluate gardens on public property. These locations would have many common landscaping species, but might also have more-consistent maintenance methods than individuals' gardens.

Fitness function The fitness function in this thesis balances two conflicting objectives, growth and water efficiency, with a user-defined weight that assigns priority to each objective. This approach is common for weighting conflicting objectives or minimizing resource use. Another option to explore in future work is to maximize growth within this constraint. For example, the optimization objective could be to maximize fitness scores such that the final arrangement uses less than 1000 mL of water.

Another option for future work on the fitness function is to explore the effects of behavioral rules for the agents that are not based solely on biological response. For example, the fitness function could be modified to reward similar plant-agent types for being close together or far

apart, or to generate clusters that include a desired number of agents, or agents with similar colors. While these types of emergent properties are often the focus of agent-based models, including non-biological features in the fitness function could present a problem in this domain. Regardless of plant color or species, the plants need to survive on the resources available at their particular location. A fitness function that rewarded non-biological properties could generate solutions with high fitness scores and “dead” plants. Nonetheless, a fitness function that includes both biological and non-biological properties presents interesting options for emergent behaviors other than water use.

Optimization The optimization routines explored in this work included two standard approaches, simulated annealing and a genetic algorithm, and a new method designed to capture the specific features of the domain. Simulated annealing modifies one item at a time and checks the effect on the global fitness score. While this method outperformed the other two routines, more complex search routines, such as tabu search, scatter search, and variable neighborhood search, could produce higher fitness scores. One problem observed here with simulated annealing was the execution time. In these optimizations, the algorithm was able to find good solutions with low cooling pressures. However, in larger, more-complex problems this may not be the case.

The agent-based search routine had the fastest runtime of the three strategies, and as such, is a good option on larger location optimization problems where slower algorithms cannot be used. To improve the fitness performance of this algorithm while still maintaining its runtime efficiency, methods for calculating local influence could be explored. Calculating the fitness scores of small groups of individuals would provide a level of control over agent interactions without the computational complexity of a global fitness calculation. Another possible modification to this algorithm could involve establishing a hierarchy among the agents, where the fitness of one species is worth more than the fitness of another species. This approach builds on the idea of local influence. A radius could be established around each high-priority plant agent and the fitness scores of all individuals within that radius could be calculated. This would establish the fitness of the high-priority individual as well as how the individual influences its surroundings.

Another avenue to continue exploring is to use the Matlab GA with different encodings. A preliminary test showed the ease with which the encoding could be changed in the Matlab GA to allow the optimization of the number and types of plants. Of the three optimization strategies used in this thesis, the GA may be uniquely suited for this type of optimization problem. Only two types of plants were used here. However, additional work could be performed using many more plant species simply by creating a unique ID for each plant type. This encoding could then be used to find the carrying capacity of a space, i.e. the number and types of plants that could grow in an area given the available resources.

Applications There are applications of this optimization framework that extend far beyond the one presented in this thesis. The model and fitness function could be modified to address other types of residential landscaping problems, such as landscape design in new construction and placement of individual species in an existing system.

Another application of this work is in the field of experimental agriculture. The area of agroforestry that advocates for placing trees amongst traditional crops to increase those crops' resilience against drought and other harsh environmental conditions. The challenge here is to find the right place for the right tree to provide the most benefit. Selecting and placing these trees is where this work could be extremely valuable. Another agricultural application for this work is optimizing placement of plants in a greenhouse. All greenhouses have areas that are warmer or receive more or less light than other areas. A system designed to capitalize on these microclimates by placing the right species in the right place could be used to determine the maximum number of plants that a greenhouse could support and thereby increase production in these environments.

Chapter 10

Conclusion

As I stated at the beginning of this thesis, residential irrigation accounts for a significant portion of residential water use. The work presented here represents an entirely new approach to this problem — a formal model for how water conservation can be achieved by optimizing the spatial arrangements on the landscape. Each of the chapters in this thesis presented the components of this approach and demonstrated their success at capturing the relevant details of the problem. The agent-based model accurately represented the interactions between plants and the landscape conditions that determine water use on residential landscapes. The fitness function was designed to balance the trade-offs between growth and water use. Results showed that water use on a landscape involves a complex balance between the water requirements of individual plants, the water that plants can access, and the competition for that water with other plants. A series of experiments in the greenhouse connected the model to real plants and validated water use and growth predictions.

The ideas presented in this thesis represent a blending of concepts from several disciplines, including computer science, operations research, plant biology, and ecology. Domain knowledge of plant ecology was needed to design the plant-growth model for each agent, and an understanding of ecology was required to present this model in the context of existing ecological modeling research. Concepts from both computer science and operations research can be found in the fitness function design and in the optimization processes. Existing metaheuristics were evaluated on this problem, including a commercially available genetic algorithm and one implemented by the thesis author. A new agent-search routine was also developed specifically for this thesis. Results from

these evaluations showed that the commercial software provides a reliable option for the location problem in this thesis, but it also requires an understanding of the problem and the software to generate meaningful solutions. The agent-search routine showed promising results: it ran faster than the other strategies evaluated.

The most significant contribution of this work, however, is not in the features that it draws from multiple disciplines, but rather, in what the combining of these elements can accomplish. The results presented in previous chapters embody a wholly new approach to water conservation — an approach that could not come from within any individual discipline alone. Crossing disciplinary boundaries created the potential for a new perspective on this problem and produced a solution that could extend well beyond an academic dissertation. Formulating landscape design as a computer science problem not only extends the leading edge of computer science in the form of a novel solution to an important problem, but also could result in novel xeriscaping recommendations for water conservation with lasting effects on Colorado and other arid and semi-arid regions.

Bibliography

- [1] E.H.L. Aarts and J.K. Lenstra. Local Search in Combinatorial Optimization. Wiley, New York, 1997.
- [2] S. Aassine and M.C. El Jai. Vegetation dynamics modelling: A method for coupling local and space dynamics. Ecological Modelling, 154:237–249, 2002.
- [3] O. Alp and E. Erkut. An efficient genetic algorithm for the p-median problem. Annals of Operations Research, 122:21–42, 2003.
- [4] T. Arentz and H. Timmermans. A multi-agent activity-based model of facility location choice and use. disP, 170(3):33–44, 2007.
- [5] R. Axtell, R. Axelrod, J.M. Epstein, and M.D. Cohen. Aligning simulation models: A case study and results. Computational and Mathematical Organization Theory, 1:123–141, 1996.
- [6] R.L. Axtell, J.M. Epstein, J.S. Dean, G.J. Gumerman, A.C. Swedlund, J. Harburger, S. Chakravarty, R. Hammond, J. Parker, and M. Parker. Population growth and collapse in a multiagent model of the Kayenta Anasazi in the Long House Valley. PNAS, 99:7275–7279, 2002.
- [7] D.A. Bailey. Commercial production of New Guinea Impatiens. North Carolina State University Cooperative Extension, 2012.
- [8] S. Bandini, S. Manzoni, and C. Simone. Dealing with space in multi-agent systems: A model for situated MAS. Proceedings of AAMAS02, 2002.
- [9] O. Barreteau, F. Bousquet, C. Millier, and J. Weber. Suitability of multi-agent simulations to study irrigated system viability: Application to case studies in the Senegal River Valley. Agricultural Systems, 80:255–275, 2004.
- [10] S. Bauer, U. Berger, H. Hildenbrandt, and V. Grimm. Cyclic dynamics in simulated plant populations. Proceedings of the Royal Society of London B, 269:2443–2450, 2002.
- [11] T. Berger, P. Schreinemachers, and J. Woelcke. Multi-agent simulation for the targeting of development policies in less-favored areas. Agricultural Systems, 88:28–43, 2006.

- [12] U. Berger and H. Hildenbrandt. A new approach to spatially explicit modelling of forest dynamics: Spacing, aging and neighborhood competition of mangrove trees. Ecological Modelling, 132:287–302, 2000.
- [13] A. Bernatzky. The contribution of trees and green spaces to a town climate. Energy Build., 5:1–10, 1982.
- [14] B. Bozkaya, S. Yanik, and S. Balcisoy. A GIS-based optimization framework for competitive multi-facility location-routing problems. Networks and Spatial Economics, 10:297–320, 2010.
- [15] R.W. Brooker, F.T. Maestre, R.M. Callaway, C.L. Lortie, L.A. Cavieres, G. Kunstler, P. Liancourt, K. Tielborger, J.M.J. Travis, F. Anthelme, C. Armas, L. Coll, E. Corcket, S. Delzon, E. Forey, Z. Kikvidze, J. Olofsson, F. Pugnaire, C.L. Quiroz, P. Saccone, K. Schiffers, M. Seifan, B. Touzard, and R. Michalet. Facilitation in plant communities: The past, present, and the future. Journal of Ecology, 96:18–34, 2008.
- [16] D.G. Brown, S.E. Page, R. Riolo, and W. Rand. Agent-based and analytical modeling to evaluate the effectiveness of greenbelts. Environmental Modeling and Software, 19:1097–1109, 2004.
- [17] R.M. Callaway. Positive interactions among plants. The Botanical Review, 61:306–349, 1995.
- [18] R.M. Callaway, R. Booker, P. Choler, Z. Kokvidze, C. Lortie, R. Michalet, L. Paolini, F. Pugnaire, B. Newingham, E. Aschehoug, C. Armas, D. Kikodze, and B. Cook. Positive interactions among alpine plants increase with stress. Nature, 417:844–848, 2002.
- [19] J.L. Casti. Would-be Worlds: How Simulation is Changing the Frontiers of Science. John Wiley and Sons, New York, 1997.
- [20] B.E. Chalker. Modelling light saturation curves for photosynthesis: An exponential function. J. Theor. Biol., 84:205–215, 1980.
- [21] F. Chiyoshi and R. Galvao. A statistical analysis of simulated annealing applied to the p-median problem. Annals of Operations Research, 96:61–74, 2000.
- [22] V. Clausnitzer and J.W. Hopmans. Simultaneous modeling of transient three-dimensional root growth and soil water flow. Plant and Soil, 164:299–314, 1994.
- [23] L.R. Costello, N. Matheny, and J. Clark. The Landscape Coefficient Method. A Guide to Estimating Irrigation Water Needs of Landscape Plantings in California. Dept. of Water Resources, Sacramento, CA, 2000.
- [24] M. Crawley, editor. Plant Ecology. Blackwell Scientific Publications, Oxford, 1997.
- [25] J. Current and M. OKelly. Locating emergency warning signals. Decision Sciences, 23:221–234, 1992.

- [26] D. Datta, K. Deb, and C.M. Fonseca. Multi-objective evolutionary algorithms for resource allocation problems. In Proceedings of the Fourth International Conference, EMO 2007. Lecture Notes in Computer Science, pages 401–416, 2007.
- [27] D.L. DeAngelis and L.J. Gross. Individual-based Models and Approaches in Ecology: Populations, Communities, and Ecosystems. Chapman and Hall, New York, 1992.
- [28] C. Dibble and P.J. Densham. Generating interesting alternatives in GIS and SDSS using genetic algorithms. In International Conference on GIS/LIS, 1993.
- [29] D. Eaton, M. Daskin, D. Simmons, B. Bulloch, and G. Jansma. Determining emergency medical service vehicle deployment in Austin, Texas. Interfaces, 15:96–108, 1985.
- [30] J. Ferber and J. Muller. Influences and reaction: A model of situated multiagent systems. Proceedings of ICMAS96, 1996.
- [31] R.L. Francis and J.A. White. Facility Layout and Location: An Analytical Approach. Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
- [32] P.R. Van Gardingen and J. Grace. Plants and wind. Advances in Botanical Research, 18:192–254, 1991.
- [33] A. Gerwitz and E.R. Page. An empirical mathematical model to describe plant root systems. Journal of Applied Ecology, 11(2):773–781, 1974.
- [34] E. Glenn, R. Tanner, S. Mendez, T. Kehret, D. Moore, J. Garcia, and C. Valdes. Growth rates, salt tolerance and water use characteristics of native and invasive riparian plants from the delta of the Colorado River, Mexico. Journal of Arid Environments, 40:281–294, 1998.
- [35] F. Glover. Heuristics for integer programming using surrogate constraints. Decision Sciences, 8:156–166, 1977.
- [36] F. Glover. Tabu search-part I. ORSA Journal on Computing, 1(3):190–206, 1989.
- [37] L. Gomez-Aparicio, R. Zamora, J.M. Gomez, J.A. Hodar, J. Castro, and E. Baraza. Applying plant facilitation to forest restoration: A meta-analysis of the use of shrubs as nurse plants. Ecological Applications, 14:1128–1138, 2004.
- [38] P. Griffin, C.R. Scherrer, and J.L. Swann. Optimization of community health center locations and service offerings with statistical need estimation. IIE Transactions, 40:880–892, 2008.
- [39] V. Grimm and S.F. Railsback. Individual-based Modeling and Ecology. Princeton University Press, Princeton, New Jersey, 2005.
- [40] E.P.M. Grist. The significance of spatio-temporal neighborhood on plant competition for light and space. Ecological Modelling, 121:63–78, 1999.

- [41] R. St. Hilaire, M.A. Arnold, D.C. Wilkerson, D.A. Devitt, B.H. Hurd, B.J. Lesikar, V.I. Lohr, C.A. Martin, G.V. McDonald, R.L. Morris, D.R. Pittenger, D.A. Shaw, and D.F. Zoldoske. Efficient water use in residential urban landscapes. HortScience, 43(7):2081–2092, 2008.
- [42] F.S. Hillier and G.J. Lieberman. Introduction to Operations Research. McGraw Hill, New York, New York, 2005.
- [43] D.S. Hochbaum. Approximation Algorithms for NP-Hard Problems. International Thomson Publishing, 1996.
- [44] R. Hoenigman, E. Bradley, and A. Lim. Cooperation in bike racing — when to work together and when to go it alone. Complexity, 2011. DOI: 10.1002/cplx.20372, Excerpted in Complexity Digest: <http://comdig.unam.mx/index.php>.
- [45] K. Hogan. Reducing errors in rainfall estimates through rain gauge location. Geographical Analysis, 22:13–49, 1990.
- [46] J.H. Holland. Adaptation in Natural and Artificial Systems. The MIT Press, Cambridge, MA, 1992.
- [47] C. Holzapfel, K. Tielborger, H. Parag, J. Kigel, and M. Sternberg. Annual plant-shrub interactions along an aridity gradient. Basic and Applied Ecology, 7:268–279, 2006.
- [48] C.M. Hosage and M.F. Goodchild. Discrete space location-allocation solutions from genetic algorithms. Annals of Operations Research, 6:35–46, 1986.
- [49] R. Hunt. Plant Growth Curves—The Functional Approach to Plant Growth Analysis. University Park Press, Baltimore, Maryland, 1982.
- [50] M. Huston, D. DeAngelis, and W. Post. New computer models unify ecological theory. BioScience, 38(10):682–691, 1988.
- [51] PlantSense Inc. <http://www.easybloom.com>.
- [52] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 28:392–403, 1998.
- [53] A.D. Jassby and T. Platt. Mathematical formulation of the relationship between photosynthesis and light for phytoplankton. Limnology and Oceanography, 21:540–547, 1976.
- [54] W. Jaziri and T. Paquet. A multi-agent model and Tabu search optimization to manage agricultural territories. GeoInformatica, 10:337–357, 2006.
- [55] S. Katare and V. Venkatasubramanian. An agent-based learning framework for modeling microbial growth. Engineering Applications of Artificial Intelligence, 14:715–726, 2001.
- [56] J. Kennedy, R. Eberhart, and Y. Shi. Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, pages 1942–1948, Perth, Australia, 1995.

- [57] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. Science. New Series, 220:671–680, 1983.
- [58] A. Klose and A. Drexler. Facility location models for distribution system design. European Journal of Operational Research, 162:4–29, 2005.
- [59] J. Knopf. The Xeriscape Flower Gardener: A Waterwise Guide for the Rocky Mountain Region. Johnson Book, Boulder, Colorado, 1991.
- [60] B. Lee and R. Deninger. Optimal locations of monitoring stations in water distribution systems. Journal of Environmental Engineering, 118:4–16, 1992.
- [61] Thomas Lemieux. personal communication, 2010.
- [62] B-L. Li, H i Wu, and G. Zou. Self-thinning rule: A causal interpretation from ecological field theory. Ecological Modelling, 132:167–173, 2000.
- [63] J. Liu, H. Jing, and Y.Y. Tang. Multi-agent oriented constraint satisfaction. Artificial Intelligence, 136:101–144, 2002.
- [64] J. Lundholm, J.S. MacIvor, Z. MacDougall, and M. Ranalli. Plant species and functional group combinations affect green roof ecosystem functions. PLoS ONE, 5, 2010.
- [65] E. Malezieux, Y. Crozat, C. Dupraz, M. Laurans, D. Makowski, H. Ozier-Lafontaine, B. Rapidel, S. de Tourdonnet, and M. Valantin-Morison. Mixing plant species in cropping systems: Concepts, tools and models. A review. Sustainable Agriculture, 29:43–62, 2009.
- [66] C.A. Martin. Landscape water use in Phoenix, Arizona. Desert Plants, 17:26–31, 2001.
- [67] C.A. Martin and L.B. Stabler. Plant gas exchange and water status in urban desert landscapes. Journal of Arid Environments, 51:235–254, 2002.
- [68] R. Mathevet, F. Bousquet, C. Le Page, and M. Antona. Agent-based simulations of interactions between duck population, farming decisions and leasing of hunting rights in the Camargue (southern France). Ecological Modelling, 165:107–126, 2003.
- [69] R.H. Merritt and H.C. Kohl. Morphology of bedding plants in response to low night temperatures and energy use implications. Scientia Horticulturae, 45:295–302, 1991.
- [70] J.A. Moreno-Perez, J.M. Moreno-Vega, and N. Mladenovic. Tabu search and simulated annealing in p-median problem. In Proceedings of the Canadian Operational Research Society Conference, Montreal, 1994.
- [71] G. Mosetti, C. Poloni, and B. Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. Journal of Wind Engineering and Industrial Aerodynamics, 51:105–116, 1994.

- [72] P. Mou, R.J. Mitchell, and R.H. Jones. Root distribution of two tree species under a heterogeneous nutrient environment. Journal of Applied Ecology, 34:645–656, 1997.
- [73] S. Moujahed, N. Gaud, and D. Meignan. A self-organizing and holonic model for optimization in multi-level location problems. In Industrial Informatics, 2007 5th IEEE International Conference on, volume 2, pages 1053 –1058, 2007.
- [74] S. Moujahed, O. Simonin, and A. Koukam. Location problems optimization by a self-organizing multiagent approach. Multiagent and Grid Systems, 5(1):59–74, 2009.
- [75] A.A. Mousa, M.A. El-Shorbagy, and W.F. Abd-El-Wahed. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. Swarm and Evolutionary Computation, page doi:10.1016/j.swevo.2011.11.005, 2011.
- [76] A.T. Murray and R.L. Church. Heuristic solution approaches to operational forest planning problems. OR Spectrum, 17:193–203, 1995.
- [77] A.T. Murray and R.L. Church. Applying simulated annealing to location-planning models. Journal of Heuristics, 2:31–53, 1996.
- [78] H.J.W. Mutsaers. Photosynthesis of cotton canopies (*Gossypium hirsutum* L.). Field Crops Research, 5:95–111, 1982.
- [79] J. Von Neumann and A.W. Burks. Theory of Self-reproducing Automata. University of Illinois Press, Urbana, Illinois, 1966.
- [80] J.M. Norman. Synthesis of canopy processes. In G. Russell, B. Marshall, and P.G. Jarvis, editors, Plant Canopies: Their Growth, Form and Function, pages 161–175. Cambridge University Press, Cambridge, 1989.
- [81] S.W. Pacala and J. Silander. Neighborhood models of plant population dynamics. I. Single species models of annuals. American Naturalist, 125:385–411, 1985.
- [82] C.H. Papadimitriou. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [83] D.C. Parker, S.M. Manson, M.A. Janssen, M.J. Hoffmann, and P. Deadman. Multi-agent systems for the simulation of land-use and land-cover change: A review. Annals of the Association of American Geographers, 93:314–337, 2003.
- [84] R.R. Pattison, G. Goldstein, and A. Ares. Growth, biomass allocation and photosynthesis of invasive and native Hawaiian rainforest species. Oecologia, 117:449–459, 1998.
- [85] M. Pogson, R. Smallwood, E. Qwarnstrom, and M. Holcombe. Formal agent-based modeling of intracellular chemical interactions. BioSystems, 85:84–100, 2004.
- [86] J.R. Porter. Modules, models and meristems. In G. Russell, B. Marshall, and P.G. Jarvis, editors, Plant canopies: Their Growth, Form and Function, pages 143–159. Cambridge University Press, Cambridge, 1989.

- [87] J.E. Preece and P.E. Read. The Biology of Horticulture: An Introductory Textbook. John Wiley and Sons, New York, 1993.
- [88] H. Pretzsch, P. Biber, and J. Dursky. The single tree-based stand simulator SILVA: Construction, application, and evaluation. Forest Ecology and Management, 162:3–21, 2002.
- [89] P. Prusinkiewicz. Modeling plant growth and development. Current Opinion in Plant Biology, 7:79–83, 2004.
- [90] C. Rademacher, C. Neuert, V. Grundmann C. Wissel, and V. Grimm. Reconstructing spatiotemporal dynamics of central European beech forests: The rule-based model BEFORE. Forest Ecology and Management, 194:349–368, 2004.
- [91] W. Rand, D.G. Brown, S.E. Page, R. Riolo, L.E. Fernandez, and M. Zellner. Statistical validation of spatial patterns in agent-based models. Proceedings, Agent Based Simulation, 2003.
- [92] C. ReVelle and R. Swain. Central facilities location. Geographical Analysis, 2:30–42, 1970.
- [93] C. Reynolds. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pages 25–34, 1987.
- [94] J.F. Reynolds and B. Acock. Predicting the response of plants to increasing carbon dioxide: A critique of plant growth models. Ecological Modelling, 29:107–129, 1985.
- [95] P.J.H. Sharpe, J. Walker, L.K. Penridge, H-I. Wu, and E.J. Rykiel. Spatial considerations in physiological models of tree growth. Tree Physiology, 2:403–421, 1986.
- [96] L. Shashua-Bar, D. Pearlmutter, and E. Erell. The cooling efficiency of urban landscape strategies in a hot dry climate. Landscape and Urban Planning, 92:179–186, 2009.
- [97] J. Shi, A. Ren, and C. Chen. Agent-based evacuation model of large public buildings under fire conditions. Automation in Construction, 18:338–347, 2009.
- [98] L. Stabler, C.A. Martin, and A.J. Brazel. Microclimates in a desert city were related to land use and vegetation index. Urban Forestry and Urban Greening, 3:137–147, 2005.
- [99] L.B. Stabler and C.A. Martin. Irrigation regimens differentially affect growth and water use efficiency of two southwest landscape plants. J. Environ. Hort., 18(2):66–70, 2000.
- [100] E. Talbi. Metaheuristics—From Design to Implementation. John Wiley, New Jersey, 2009.
- [101] K. Tielborger and R. Kadmon. Temporal environmental variation tips the balance between facilitation and interference in desert plants. Ecology, 81:1544–1553, 2000.
- [102] J.M.J. Travis, R.W. Booker, and C. Dytham. The interplay of positive and negative species interactions across an environmental gradient: Insights from an individual-based simulation model. Biol. Letters, 1:5–8, 2005.

- [103] N. Ulder, E. Aarts, H-J. Bandelt, P. van Laarhoven, and E. Pesch. Genetic local search algorithms for the traveling salesman problem. In Hans-Paul Schwefel and Reinhard Manner, editors, Parallel Problem Solving from Nature, volume 496 of Lecture Notes in Computer Science, pages 109–116. Springer Berlin / Heidelberg, 1991.
- [104] D.C. Walker, J. Southgate, G. Hill, M. Holcombe, D.R. Hose, S.M. Wood, S. MacNeil, and R.H. Smallwood. The epitheliome: Agent-based modelling of the social behaviour of cells. BioSystems, 76:89–100, 2004.
- [105] J. Walker, P.J.H. Sharpe, L.K. Penridge, and H-I. Wu. Ecological field theory: The concept and field tests. Vegetatio, 83:81–95, 1989.
- [106] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer. Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. IEEE Transactions of Evolutionary Computation, 7:189–203, 2003.
- [107] J. Weiner. A neighborhood model of annual-plant interference. Ecology, 63(5):1237–1241, 1982.
- [108] J. Weiner, P. Stoll, H. Muller-Landau, and A. Jasentuliyana. The effects of density, spatial pattern, and competitive symmetry on size variation in simulated plant populations. The American Naturalist, 158(4):438–450, 2001.
- [109] N.T. Welander and B. Ottosson. Influence of photosynthetic photon flux density on growth and transpiration in seedlings of *Fagus sylvatica*. Tree Physiology, 17:133–140, 1997.
- [110] N.T. Welander and B. Ottosson. The influence of low light, drought and fertilization on transpiration and growth in young seedlings of *Quercus robur* L. Forest Ecology and Management, 127:139–151, 2000.
- [111] D. Whiting, M. Roll, and L. Vickerman. Plant Growth Factors: Light – Garden Notes 142. Colorado State University Cooperative Extension, 2007.
- [112] M. Woolridge. Intelligent agents. In G. Weiss, editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pages 27–78. MIT Press, Cambridge, MA, 1999.
- [113] C. Wu, Q. Xiao, and G. McPherson. A method for locating potential tree-planting sites in urban areas: A case study of Los Angeles, USA. Urban Forestry and Urban Greening, 7:65–76, 2008.
- [114] H-I. Wu, P.J.H. Sharpe, J. Walker, and L.K. Penridge. Ecological field theory: A spatial analysis of resource interference among plants. Ecological Modelling, 29:215–243, 1985.