

A Method for Optimizing a Metapopulation

Fabian Grüning, Michael Sonnenschein

Carl von Ossietzky Universität Oldenburg
Department of Computing Science, Environmental Informatics Group
D-26111 Oldenburg

Abstract: Metapopulations are a popular way to gain an abstract model of a certain species in a scattered landscape. These models can be used for analytical questions concerning the viability of the modeled species. Several applications exist that evaluate metapopulations concerning their persistence, e.g. Meta-X.

There is a strong interest in discovering modifications of the environment, that have a positive influence in the survivability of the population. Recommendations for those modifications, that can be evaluated by tools like Meta-X, are mainly given from experts. Automatically generated recommendations were hardly available mainly because of the high complexity of the problem.

This paper shows the results of an implementation of an optimization algorithm for methods to increase viability of metapopulations. It uses a metamodel of a metapopulation model for accomplishing the needed performance.

Keywords: Planning for environmental engineering (EE), Ecological modelling

1 Introduction

This work is based on the metapopulation model provided by Meta-X, which is a tool for metapopulation viability analysis [Frank⁺03]. This metapopulation model is used for creating an ecological model of a certain landscape that can be simulated and analysed regarding its viability. A metapopulation consists of patches possibly colonized by a population of a specific species. Patches can be linked to each other so that individuals can leave one patch and arrive at another one. Patches can be recolonized through this process if they have gone extinct due to environmental or demographic reasons.

For the protection of species it may become necessary to improve the environmental characteristics of the modelled environment. To use the available resources optimally an algorithm is designed that calculates a distribution of costs that leads to a high viability.

Some characteristics of the metapopulation may be changed by human efforts to increase the metapopulation's viability. In that case, a certain amount of costs is applied to those characteristics with intend to maximize the viability. It is a difficult task to provide the best allocation of the costs, which is taken over by the optimization algorithm proposed in this paper. This algorithm is an intelligent search

method for optimal allocation of costs to specific metapopulation's characteristics. Because of the high complexity of the optimization problem, runtime is a crucial factor. To compute the desired allocation as fast as possible, the most time consuming aspect is by-passed by optimizing a functional metamodel of the metapopulation model instead of the simulation model itself.

Without this procedure, the optimization process would not be practicable. Furthermore, the generation of the metamodel is implemented parallelly for another speed up.

The optimization process to maximize a metapopulation's viability with respect to cost restrictions is split up into three tasks, which will be explained in detail below:

1. generating a training set of metamodel's configurations and their viability from the Meta-X simulation model,
2. training of a metamodel that maps metamodel's configurations onto their viability using support vector machines,
3. optimizing the metamodel by applying a certain amount of costs to its characteristics so that the viability reaches its maximum.

The implementation takes a metapopulation model designed and exported from Meta-X as input and produces an optimized model in a Meta-X importable format. As Meta-X provides a simple point-and-click interface for generating metapopulation models, these two programs complement one another when it comes to automatic optimization of the specified metapopulations.

Other approaches for optimizing metapopulation models can be found in [CaMo03], [MoCa02] and [Cabeza⁺04]. These approaches can also be used for planning concrete modifications in the environment for the protection of species, but they differ regarding to the way they optimize a given metapopulation: They find a set of reserves that provide a high rate of biodiversity, especially taking in consideration the loss of the unselected reserves. These algorithms make use of different techniques to gain their results than the method described in this paper and do not use a metamodel.

2 The metapopulation

This is a very short introduction of the metapopulation model of Meta-X, that is described in greater detail in [Frank⁺03]. The topics that are covered here are the ones that are needed for understanding the optimization algorithm but not the whole metamodel.

2.1 The metamodel

A metapopulation consists of several patches that have a position in space, can be occupied or empty and may have connections to other patches. Additionally patch i has

- a local rate of extinction v_i
- an amount of individuals leaving the patch in a time period E_i
- a needed amount of individuals for recolonization of the patch in case the patch has gone extinct I_i

After a short amount of time Δt , one of three possible events changes the state of the metapopulation. These events can be

- Local extinction: Every patch i has a short term probability to go extinct due to intrinsic patch dynamics,
- Correlated extinction: Two patches may become extinct at the same time because of regional changes of the environment,
- Recolonization: If a patch j is currently empty and a patch i is colonized and those two patches are connected to each other, patch j may get recolonized due to individuals that populate the empty patch.

The metapopulation becomes extinct, if the last patch is empty and lasted for the sum of the Δt s that passed by.

2.2 Cost functions

Many of the parameters the metapopulation model consists of cannot be altered, because they are inherent to the examined population or environment. However, two parameters can be improved by applying costs to them (see below).

There has to be a function that maps the costs applied to the parameters of the metapopulation model in a way, that the parameter settings cannot become invalid independent from the amount of costs applied to them. Furthermore it is welcomed, that the marginal productivity is decreasing, i.e. further costs applied to a parameter have less effect.

The cost functions given below satisfy these requirements and are therefore considered to be adequate. As there are no possibilities for an empirical verification of these functions they remain artificially chosen.

2.2.1 Modifiable parameters

The following two parameters can be improved by changing environmental conditions. These changes can be expressed by applying costs to the parameters of

the metapopulation, that have their counterparts in reality in building fences, relocating streets, etc.

- v_i is the local rate of extinction that can be decreased by improvements of the patches characteristics that are important for the examined population.
- d_{ij} cannot directly be manipulated as the patches have a certain position in space. But as the recolonization is strongly connected to the distance between two patches, it can be shortened for the individuals of the examined population by carrying out modifications to the environment, that ease the moving for the individuals from patch i to patch j so that more empty patches can be recolonized.

2.2.2 Sigmoid Optimization

v_i is, as the local extinction rate, a probability and is therefore valid in $[0;1]$. The $c_S(x)$ is the cost function that is used to apply costs to the local extinction rates in order to improve the survival rate of the population in patch i .

$$c_S(x) = 1 - 2 \cdot \left(\frac{1}{1 + \exp(-ax)} - \frac{1}{2} \right) \quad (6)$$

It is important to mention that the survival rate increases when the rate of extinction decreases.

In (6) is x the sum of the value of the inverse function of $c_S(x)$ of the old value for the local extinction rate and the costs that are applied to it. Then the value of (6) is the new local extinction rate.

The coefficient a is used to assess the applied costs to the local extinction rate in reference to the costs applied to the distances of the patches (see (7)).

2.2.3 Exponential Optimization

The distance between the patches i and j can be decreased in the sense that the examined individuals have less difficulties to move from one to the other. As valid values for the distances between patches are in \mathbb{R}^+ , $c_E(x)$ is used to apply costs to the distances of the patches in order to decrease them.

$$c_E(x) = \begin{cases} \exp(bx) - 1 & \text{if } \exp(bx) - 1 > 0 \\ x & \text{else} \end{cases} \quad (7)$$

The value for the new patch distance is calculated in the same way as the value for the local extinction rate in 2.2.2, but the costs are negative because lower distances mean better reachability.

The coefficient b is the counterpart to the coefficient a in (6). These coeffi-

cients can be used to control the intensity of the effects the applied costs have on their respective values.

3 Simulation algorithm

The simulation algorithm is used to compute the viability of a modeled metapopulation. To measure the viability the mean time to extinction is used. The higher the value the longer the metapopulation will last.

This value is an artificial value calculated from the extinction times of a number of concrete simulations of the given metapopulation model. It can only be used to compare two metapopulations and to bring them into a sequence. With this tool, one can evaluate the efficiency of two different distributions of costs applied to the metapopulation model's parameters according to the survival rate of the population.

The Meta-X model uses stochastic time steps, which means that the time between every two events is not uniform. Two actions determine a changing in the model's state:

1. determination of the transition time the metapopulation lasts in the current state,
2. determination of the transition event that takes place.

Again, the complete algorithm can be found in [Frank⁺03].

3.1 Calculating the mean time to extinction

The mean time to extinction can now be calculated from many extinction times. The procedure is a modified version of [Stelter⁺97, pp. 508] that allows to use continuous time. We utilize that the extinction times P_0 , that are gained from the simulations and calculated by the sum of the Δt s that passed by until the population dies (see chapter 2.1), are exponentially distributed:

$$P_0(T) = 1 - \exp(-T/T_m) \quad (10)$$

where T is the simulation time and T_m is the mean time to extinction that we are looking for. With (10) we can formulate the linear equation

$$-\log(1 - P_0(T)) = \frac{1}{T_m} T \quad (11)$$

which can be identified through linear regression. After that, the mean time to extinction is calculated by the reciprocal value of the gradient of (11).

3.2 Performance issues

The mean time to extinction of a metapopulation is calculated using the extinction times from the metapopulations simulations. For statistical reasons, a lot of simulations, e.g. 1000, have to be executed, which is quite a high requirement of computational power. Therefore the simulations should be executed parallelly on a multi-processor machine or in a network of independent machines so that every processor or machine calculates one simulation at a time. The increase of speed is nearly linear regarding the attached machines so this is a good way to speed up the process. As this is the most critical part in the process regarding computational power, it is recommended to use as many machines as possible.

4 The metamodel

The metamodel is used to speed up the optimization process. Instead of optimizing a metapopulation by its Meta-X model directly, a metamodel is built and used for optimization instead.

As shown in chapter 3, the process to gain the mean time to extinction from a metapopulation model is a time consuming computation. As the optimization algorithm needs the resulting mean time to extinction for many configurations of the metapopulation, it is cheaper, regarding computational time, to build a training set of randomly improved metapopulation models and their mean time to extinctions and to train a metamodel by using that training set, than to run the algorithm for computing the mean time to extinction every time the optimization algorithm needs the result.

The results of the metamodel are not exactly the same as for Meta-X models, because the algorithm approximates the desired value. However, as alone through the remarkable gain of speed by using the metamodel makes the computation of the optimized metapopulation possible, the loss of precision is acceptable and tests show, that the results of the optimization process are very good [Grüning05].

The concrete implementation uses an API from [ChLi01] for building the metamodel.

4.1 Support vector machines

Support vector machines are used to produce a metamodel from a training set of example mappings of an unknown function. In this case the metapopulation's configuration, which parameters got randomly improved by applying costs, is the domain and the resulting mean time to extinction is the range of the function. The function, which maps the improved metapopulations to their mean time to extinction, is unknown, but the algorithm will provide an approximation of the function through adaptation of the examples in the training set. By using this technique, the

values for the mean time to extinction will be delivered much faster than by using the simulation algorithm (see chapter 3).

An example, taken from [RuNo03, pp.749], will show how the support vector machines work. This is not an exhaustive discussion of this technique, but it will give an impression of how support vector machines work. More about support vector machines can be found at [Sc⁺00; Sc98, pp. 135; SmSc98].

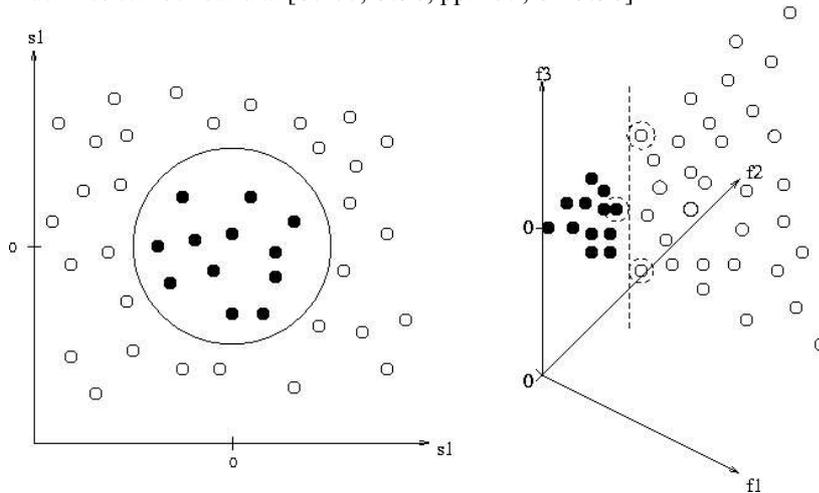


Figure 1: The left side shows the training set containing black and white dots, where the black dots are inside a circle so that the two types of dots cannot be linearly separated. The right side shows the linearly separable set after applying the kernel function. The support vectors are additionally surrounded by circles.

4.2 Mapping the input

We start in a two dimensional space, where we have attributes $s = (s_1, s_2)$ with two different examples $y = \pm 1$. One of them lie in a circular area (see figure 1 on the left side), which means that the two sets are not linearly separable.

We now use a function $F(s) = (f_1, f_2, f_3)$ that maps each input to a new vector by using the features

$$f_1 = s_1^2, f_2 = s_2^2, f_3 = \sqrt{2} s_1 s_2. \quad (12)$$

$F(s)$ maps the inputs to a three dimensional room, where the two different classes of examples are linearly separable (see figure 1 on the right side). This effect can be produced in general by mapping the inputs to a sufficiently high dimensional space. But by using a too high dimensional space the problem of overfitting appears, which means that nearly every input could be mapped to its own dimension becoming its own class.

4.3 Support vectors

To not run into that trap, an optimization algorithm is used to find the optimal linear separator, which means that the margin between the different examples is maximal. This problem can be solved efficiently through quadratic programming by finding the values of the α_i s that maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (13)$$

under the constraints $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$, where the examples x_i and x_j have the classes $y_i = \pm 1$.

After the values for α_i s have been determined, which are unique, the class of the example x can be found by

$$h(x) = \text{sign} \left(\sum_i \alpha_i y_i (x \cdot x_i) \right) \quad (14)$$

The examples next to the border of each class are called support vectors (see figure 1 on the right side). Only the α_i s belonging to the support vectors are non-zero.

4.4 Kernels

We now need to find a linear separator in a space with enough dimensions. This is done by replacing $x_i \cdot x_j$ in (13) with $F(x_i) \cdot F(x_j)$.

This replacement has the pleasant characteristic that by involving (12)

$$F(x_i) \cdot F(x_j) = (x_i \cdot x_j)^2 \quad (15)$$

This means that $F(x_i) \cdot F(x_j)$ can be computed without computing F for each point.

$(x_i \cdot x_j)^2$ is called the kernel function and can often be found as $K(x_i, x_j)$. There are many kernel functions that can be used, as long as it evaluates the dot product of two input samples, where the kernel used to build the metamodel for the metapopulations is the radial basis function

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (16)$$

where γ is a kernel parameter.

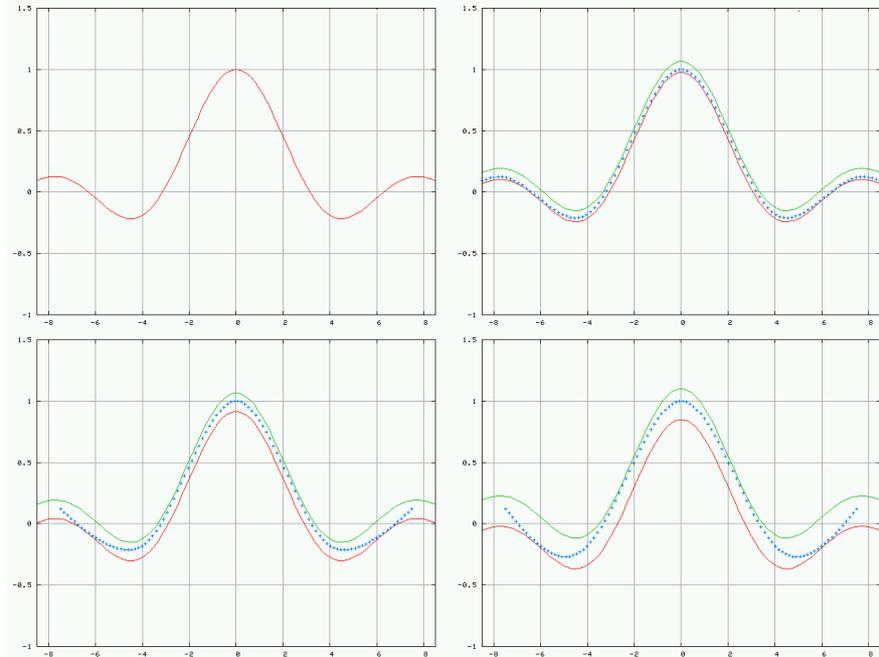


Figure 2: Interpolation polynomial (blue crosses) for increasing errors, which leads to increasing radii. The first picture shows the function that has to be approximated. The other pictures show the resulting interpolation polynomial for different error values.

4.5 Interpolation

The support vectors are used for the Lagrange polynomial interpolation [Späth94], which provides the desired functional approximation of the training set in our application. The amount of support vectors used for the interpolation depends on a chosen error ϵ .

The error ϵ defines the radius of a pipe that encloses the function that has to be approximated, in which the interpolation polynomial must be located. Figure 2 shows the interpolation polynomial for different ϵ s.

If the error decreases, more support vectors are used to specify the interpolation polynomial. Figure 3 shows the dependency between the support vectors and the resulting interpolation.

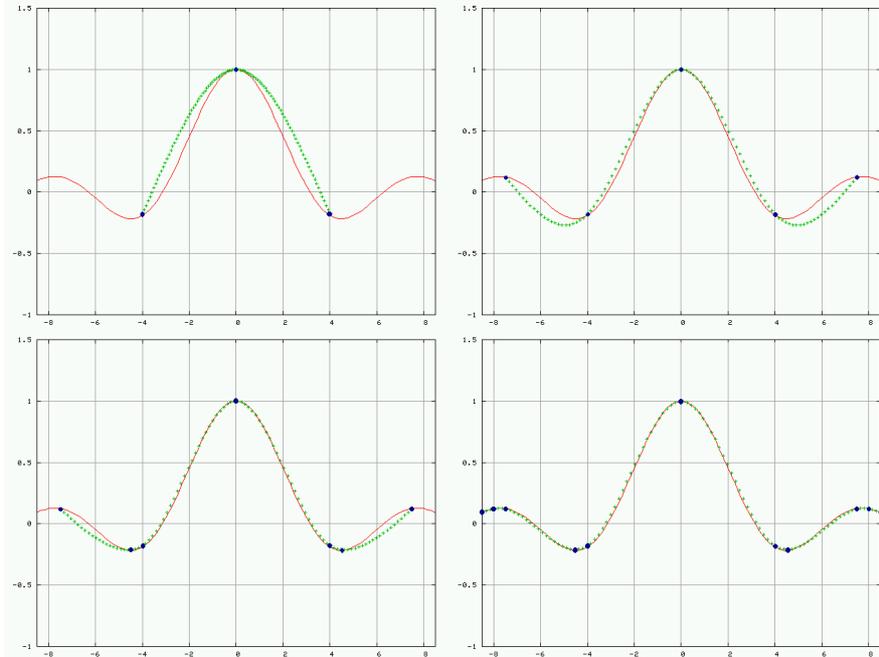


Figure 3: The error of the interpolation polynomial (function of crosses) becomes lesser the more support vectors (blue dots) are used for the Lagrange interpolation algorithm. The error determines the quality of the interpolation polynomial.

5 Optimization algorithm

The optimization algorithm is used to apply costs to the metapopulation model's parameters in that way, that the mean time to extinction reaches its possible maximum for the given costs. There are several algorithm that can be used to achieve this; the one that is used here is tabu search [MiFo00].

The tabu search algorithm does not optimize the metapopulation model itself but the metamodel that was built from the metapopulation's samples. The optimization algorithm needs numerous values of the mean time to extinction for certain metapopulations. The metamodel will provide the answer much faster than it would the metapopulation model itself by using its simulation algorithm to compute the mean time to extinction.

The optimum computed by the metamodel has not necessarily to be the optimum for the metapopulation. But the bigger the training set for building the metamodel is, the better will the metamodel become and the nearer is the resulting optimum of the metamodel to the metapopulation's optimum.

5.1 Hill climbing

The tabu search basically uses hill climbing. The hill climbing algorithm starts from the original metapopulation configuration, where randomly costs have been applied to it, and steps from metapopulation configuration to metapopulation configuration until it reaches a maximum and stops. Therefor for each step it analysis the neighboring metapopulation configurations and valuates them regarding to their mean time to extinction. It then chooses the best configuration and makes the move.

5.2 Tabu list

The hill climbing algorithm has the disadvantage that it cannot escape form local maxima, because no neighboring metapopulation offers a higher mean time to extinction. To achieve better results a tabu list is used, where the last n steps are remembered. The algorithm is not allowed to step on a metapopulation's configuration, which is in that list. Additionally the algorithm is now forced to make a step, even if it leads it to a configuration of a lower mean time to extinction.

The behaviour of the algorithm now changes in that way, that it can escape from local maxima. After reaching a local maximum, the algorithm is forced to go downhill, because the way uphill is taboo. It then can reach another maximum.

The algorithm stops after a certain amount of steps and returns the best metapopulation found so far.

6 Example of an optimization

An example of use will show the capabilities of the optimization algorithm by optimizing a concrete metapopulation model, that is created by Meta-X. The results of the optimization process are then compared to the original scenario and dicussed regarding their effectivity by comparing them to standard ecologist's procedures.

6.1 The scenario

The scenario consists of six patches that are divided by a road so that three patches are on the one side and the other three patches are on the other side of the road. The road represents a barrier for the individuals of the species so that they have a high possibility to die by trying to move from a patch from the one side of the road to a patch from the other side of the road. This scenario's characteristic is designed by having a huge distance between the patches that lie on the different sides of the road so that the possibility to move successfully to the other side of the road is low. Figure 4 shows a graphical representation of the scenario.

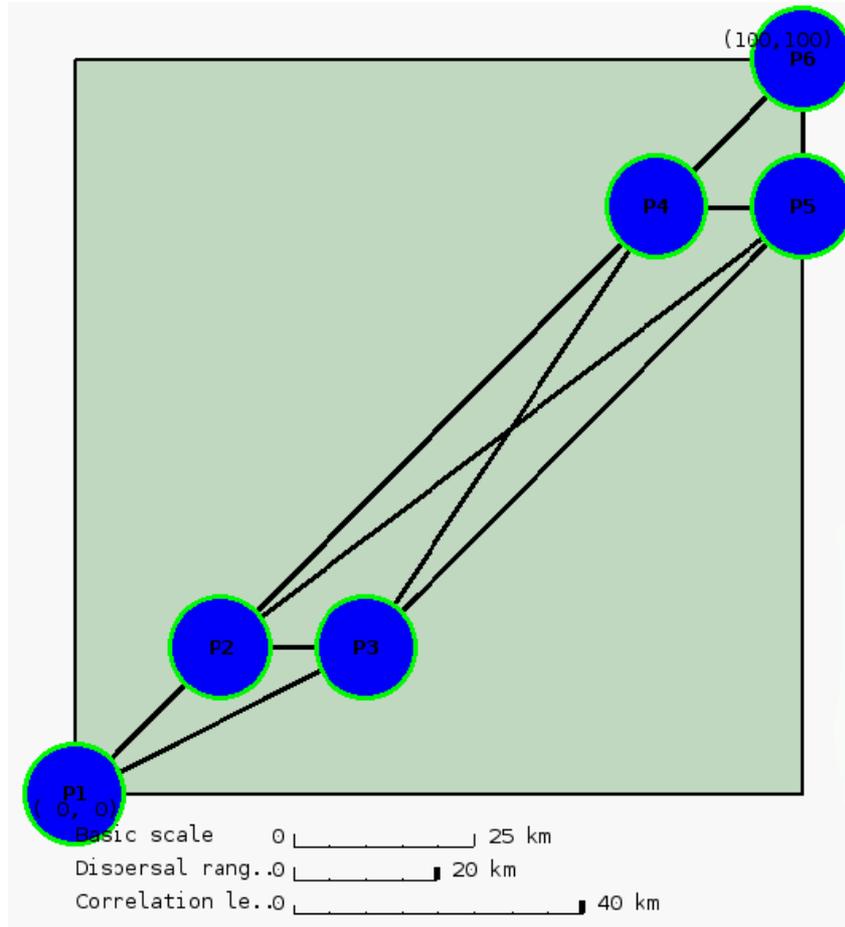


Figure 4: The scenario to be optimized consists of six patches, where the patches 1-3 are on the one side and the patches 4-6 are on the other side of the road. The patches are only connected to their direct neighbors.

6.2 Optimization's results

The optimization algorithm applies two units of costs to the local extinction rate and the patch distance of the metapopulation model. Tables 1 and 2 show the resulting values of the metapopulation model's characteristics.

The mean time to extinction calculated by Meta-X gets advanced from 118 to 135, which is an improvement of 15%.

Patch	Before	After
P1	0.019999862	0.016567826
P2	0.049999833	0.049999833

Patch	Before	After
P3	0.03999996	0.035209537
P4	0.029999971	0.029999971
P5	0.03999996	0.03452456
P6	0.019999862	0.018660545

Table 1: The local extinction of the patches before and after the application of costs by the optimization algorithm.

Patch1	Patch2	Before	After
P1	P2	28.28	28.28
P1	P3	44.719997	25.115685
P1	P4	113.14	113.14
P1	P5	128.06	128.06
P1	P6	141.42	141.42
P2	P3	20.000002	20.000002
P2	P4	84.84998	84.84998
P2	P5	99.99999	67.382744
P2	P6	113.14	113.14
P3	P4	72.109985	72.109985
P3	P5	84.84998	84.84998
P3	P6	100.0	100.0
P4	P5	20.000002	17.625334
P4	P6	28.28	20.475328
P5	P6	20.000002	18.192558

Table 2: The patch distances before and after the optimization algorithm applied costs on them.

6.3 Discussion of the results

First of all the mean time to extinction rises, which is the desired effect for the optimization algorithm. The allocation of the costs shows, that some costs were applied to improve the local extinction rates of some patches and most of the costs were used to improve the possibility of successfully moving between the patches P2 and P5.

This distribution of costs should be understood as a priority list where to begin to improve the environment. In this case it clearly says to improve the possibility for the individuals to move between the patches P2 and P5. As already said before, the costs are not meant as an amount of real money and so the distribution of the costs the optimization algorithm recommends should not be slavishly obeyed.

This way to apply costs to a metapopulation model means to build a green bridge between these two patches and is a well known procedure by ecologists to increase the viability of a population. The counterpart in reality could be pipes beneath the road ease the way between the patches for small animals like frogs and hedgehogs. Small fences sideways at the rest of the road guide the way towards the bridges. The procedure depends on the examined species.

Other improvements would be thinkable, like easing every possible crossing of the road which would have been suggested by an equal distribution of the costs to each distance between the patches that lie on different sides of the road and would have been executed by lowering the speed limit of the road. But the optimization algorithm has found the green bridge to be more effective.

The optimization algorithm has made up an approved way to improve the viability of a population, and it gives direction where exactly to apply this method in the concrete metapopulation model, so that the resulting mean time to extinction reaches its maximum for the available amount of costs.

7 Conclusion

Although metapopulations are an often used way to gain knowledge about a certain species in a landscape, there was not a convenient way to automatically optimize those landscapes because of the computational complexity of the problem. Experts thought improvements for the given situation up, which were weight up against each other by the results of the improved metapopulation's simulations so that a decision could be made.

Through the procedure suggested by this paper it is now possible to systematically walk through the solution space of the possible applications of costs to the metapopulation's parameters to gain the distribution that maximizes the utility for the metapopulation.

This procedure does not replace the knowledge of real experts, but it can give impulses in what direction to think when it comes to optimize certain metapopulations that may be complex or extraordinary. The algorithm has the advantage of reviewing a large set of possible solutions, so that an uncommon solution may be found that a human might overlook.

The suggested procedure is more than a toy example and can probably be adopted to many other population models than Meta-X, where it already works [Grüning05]. It seems to be a convenient way of gaining knowledge on how to improve a certain landscape and can be used by ecologists and decision makers.

References

- [Frank⁰³] Frank, K.; Lorek, H.; Köster, F.; Sonnenschein, M.; Wissel, C.; Grimm, V.: Meta-X. Springer-Verlag Berlin Heidelberg, 2003.
- [CaMo03] Cabeza, M.; Moilanen, A.: Site-Selection Algorithms and Habitat Loss. *Conservation Biology*, pp. 1402-1413, 2003.
- [MoCa02] Moilanen, A.; Cabeza, M.: Single-Species dynamic Site Selection. *Ecological Applications*, pp. 913-926, 2002.
- [Cabeza⁰⁴] Cabeza, M.; Araújo, M.B.; Wilson, R.J.; Thomas, C.D.; Cowley, M.J.R.; Moilanen, A.: Combining probabilities of occurrence with spatial reserve design. *Journal of Applied Ecology*, 2004.
- [Stelter⁹⁷] Stelter, C.; Reich, M.; Grimm, V.; Wissel, C.: Modelling persistence in dynamic landscapes: lessons from a metapopulation of the grasshopper *Bryodemella tuberculata*. *J Anim Ecol* 66, 1997.
- [ChLi01] Chang, C.-C.; Lin, C.-J.: LIBSVM : a library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [RuNo03] Russel, S.; Norvig, P.: *Artificial Intelligence - A modern Approach*. Prentice Hall, 2003.
- [Sc⁰⁰] Schölkopf, B.; Smola, A.; Williamson, R.; Bartlett, P. L.: New support vector algorithms. *NeuroCOLT Technical Report NC-TR-98-031*, Royal Holloway College, University of London, UK, 1998. *Neural Computation* 12(5):1207-1245, 2000. <http://www.kernelmachines.org/papers/tr-31-1998.ps.gz>, 2004.
- [Sc98] Schölkopf, B.: *Support-vektor-lernen*. G. Hotz et al., editor, *Ausgezeichnete Informatikdissertationen*. Teubner, Stuttgart, 1998. <http://www.kernel-machines.org/papers/beitrag.ps.gz>, 2004.
- [SmSc98] Smola, Alex J.; Schölkopf, Bernhard: *A Tutorial on Support Vector Regression*. *NeuroCOLT2 Technical Report Series NC2-TR-1998-030*, 1998. <http://www.kernel-machines.org/papers/tr-30-1998.ps.gz>, 2004.
- [Späth94] Späth, Helmuth: *Numerik*. Vieweg, 1994
- [MiFo00] Michalewicz, Z.; Fogel, D. B.: *How to Solve It: Modern Heuristics*. Springer, 2000.
- [Grüning05] Grüning, F.: *Algorithmische Optimierung von Metapopulationsmodellen*. Diplomarbeit in Informatik, Oldenburg, 2005.