**Introduction**

This document describes the subcomponents for the optimization inside the site-selection procedure of the ABM model of the MEGADAPT project. This procedure searches for the set of spatial units for which a socio-institutional agent may undertake actions to maximize the return. The agent is subjected to budgetary constraints. The selection of spatial alternatives is achieved through a combination of non-dominated sorting and a genetic search algorithm. Non-dominated sorting produces a sub-population of spatial units based on the ordering of their Pareto dominance. Using this sub-population, the genetic search algorithm identifies the “best” set of actions to be taken on different spatial alternatives.

The document describes the NetLogo procedures in the sequential order shown in Figure 1.

**Definitions:**

Strategy: Strategy is the combination of actions on all spatial units.

Particle: The genetic search algorithm begins with a set of particles. Each particle is a strategy.

Fitness: Each strategy is represented as a string of numbers (0s and 1s). After each round of simulation, the idea is to delete the *n* worst particles, and to breed *n* new ones from the best particles. Each particle, therefore, needs to be awarded a figure of merit, and this is generated by the fitness function.

Mutation: Mutation is the part of the site-selection procedure, which is related to the exploration of the “best” strategy. Mutation occurs to maintain diversity within the particles and prevent premature convergence of the search algorithm. Two types of mutations are performed in this genetic search algorithm: horizontal and vertical mutation. These are explained later in the document.

**Obtain normalized criteria values:**

The values of the distance metric for each spatial alternative and action are obtained from the “Site-suitability” procedure. The values for each action are stored in separate lists that we call .

**Non-dominated sort:**

In order to identify the set of spatial alternatives more suitable for being selected for investments in action, the non-dominated sorting genetic algorithm (Deb et al., 2002) is used. The objective of this algorithm is to reduce the alternatives by eliminating the ones that would never be selected due to the low dominance. The *doNondominatedSorting* function inside the *ecr* package in *R* performs a non-dominated sorting that ranks the alternatives. The *doNondominatedSorting* function requires a numeric matrix,, as input:

Subject to the budget constraint, the non-dominated sorting algorithm reduces the dimensionality of the total spatial alternatives, and the function returns a vector of ranks,, indexed in increasing order with

The NetLogo procedure for the non-dominated sorting genetic algorithm is called the *doNondominatedSorting*.

**Initialize Random Population**

An initial population of particles of size *Particles-population-size* is first created. *Particles-population-size* is a model parameter that is set by the user. Each particle, *s*, is composed of a strategy, *Gs*.

where.

The method for generating includes two steps. In the first step, a matrix of size is generated with zeroes:

In the second step, ones are assigned to different positions in using the *random* command in NetLogo. The allocation of ones is constrained by the number of actions possible on a census block (=1):

The NetLogo procedures for initializing the particle population are *setup\_InitialParticles, setup-genome,* and *setup-Particles.*

**Calculate fitness**

This procedure calculates a fitness matrix, which is defined as the product of each decision in the landscape, and the normalized criteria value to each action, . The fitness matrix, , is calculated through matrix multiplication.

where

The total fitness of each solution, is the row and column summation of the fitness matrix.

The NetLogo procedure for calculating the fitness is called *setup-Particles.*

**Replication**

The replication procedure involves two actions: reproduction of solutions with high fitness and discarding of solutions with low fitness.

Once the total fitness of all the particles have been evaluated, particle set containing *mutant-size* number of particles with the highest total fitness are used for reproduction using the *hatch* function in NetLogo. *mutant-size* is a model parameter that is set by a slider. The *hatch* function generates a new particle with the same fitness as the parent particle.

Particle set containing *mutant-size* number of particles with the lowest total fitness are discarded using the *die* function in NetLogo. Equal number of particles are discarded and reproduced to maintain the same total particle population.

The NetLogo procedure for reproduction and omitting chromosomes is called *replicate-Particles.*

**Horizontal Mutation**

The horizontal mutation procedure is performed over the particles that newly created in the *replicate-*Particles procedure. In this operator, two columns within the matrix of the new particle population are selected at random and swapped (Fig 2). Then, the total fitness of the particles is recalculated.

The NetLogo procedure for horizontal mutation operator is called *mutate\_horizontalstrategies.*

**Vertical Mutation**

The vertical mutation procedure is also performed on the particles that are newly created in the *replicate-Particles* procedure. In this operator, two rows within the matrix of the new particle population are selected at random and swapped (Fig 3). Then, the total fitness of the particles is recalculated.

The NetLogo procedure for vertical mutation operator is called *mutate\_verticalstrategies.*

**Stop Criterion**

The *replication, horizontal mutation,* and *vertical mutation* procedures are repeated for 10,000 runs and the optimal strategy for allocation of actions on spatial units is reported.

The NetLogo procedure for checking the stopping criterion is called *optimization.*

**Reference**

Deb, K., Pratap, A., and Agarwal, S. *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6 (8) (2002), 182-197