Supplementary Material 1: Model description

This supplementary material to the paper, "Effects of stakeholder empowerment on crane population and agricultural production" provides the model description following the ODD (Overview, Design concepts, Details) protocol for describing individual- and agent-based models (Grimm et al., 2006), as updated by Grimm et al., (2020). This material covers the full ODD description including the Details sections. The Overview and Design concepts can also be found in the main document.

1. Purpose and patterns

The purpose of the model is to predict how increasing the ability of individual stakeholders to enact decisions at the farm scale affects broader scale changes in expected crane population sizes and agricultural production in four possible management scenarios over 30 years. To do this, we used the package GMSE v0.6.0.4 in R (Duthie et al., 2018). GMSE simulates the management strategy evaluation process in a way that models goal-oriented behavior and spatial distribution of individual stakeholders, managers and wildlife using an individual-based (*i.e.*, agent-based) framework (Bunnefeld et al., 2011; Duthie et al., 2018). The four management scenarios simulated were: *a.*) no management and no stakeholder power to affect cranes, *b.*) scaring and culling of cranes, with a management objective to allow the population to increase to an effectively high management target (*i.e.*, 100,000 ind.), *c.*) only culling allowed, but with an effectively high management target and *d.*) scaring and culling with a management objective to keep the population at a lower target (*i.e.*, 15,000 ind.) to lower the negative impact on agricultural production.

Management scenario A
Scenario *a* is a null model for population size and mean agricultural production when no scaring or culling is conducted and the crane population can grow exponentially. Currently, there is a lack of management targets both nationally and for the whole population along the flyways, so this reflects the current lack of population regulation.

Management scenario B
Scenario *b* models current management in Sweden, which allows for intensive scaring methods to reduce loss in agricultural production (specifically using scaring by the use of gas cannons, scare crows, flags). Very limited culling (*i.e.*, up to 15 individuals licensed to a minority of the farmers) is occasionally permitted by the managing authorities (EEA, n.d.). In our model, the strict licensing procedure for culling is represented by a high cost for culling for the farmer, unlike scaring, and was regulated in the simulations by setting a very high management target, *i.e.*, ensuring a condition where the manager would always set a high cost for culling. Non-lethal scaring may divert cranes from single cells at the landscape, but will not affect population size and a scenario only permitting scaring was therefore not considered.

Management scenario C
Scenario *c* models a hypothetical scenario where although culling is expected to be costly, thus the manager is expected to keep costs of culling very high) there either is no alternative action possible, or alternative actions like scaring are prohibited or widely perceived to be ineffective and therefore never taken.

Management scenario D
Scenario *d* models a situation where managers permit both scaring and culling, but managers attempt to keep the crane population at a size well below presumed carrying capacity. This allows for a more dynamic allocation of budget for both farmers and managers, which consequently mimics a trade-off between the objectives to sustain agricultural production while ensuring viability of the crane population.

We evaluate our model by its ability to reproduce the pattern of population growth in the previous 30 years leading up to the current population year 2020. By parameterizing the model with the parameter values (*i.e.*, initial population size and reproductive output, for details see '*Initialization*' below) based on previous empirical data on crane numbers, we can test if the model reproduces the pattern of population growth given realistic initial conditions. Recovering this pattern demonstrates that the model was fit for purpose in that we could then use it to test new scenarios against (*i.e.*, management scenario *a*-*d*).

2. Entities, state variables and scales

Farmers, managers and cranes operate on an agricultural landscape (L), modelled as a torus of discrete cells owned by individual farmers and producing a potentially variable yield of agricultural crop (*i.e.*, agricultural production). Each landscape cell thus has unique traits including farmer ownership, *x*-*y* location (L_{xy}), agricultural production and density of cranes in each time step. Only one farmer can own a single L_{xy} cell, but any number of cranes can occupy a single cell. In this study, L was constructed as a grid of 100 × 100 cells, representing a total staging area of about 200 km^2 (Nilsson et al., 2018), utilized by a total of 50 farmers earning the majority of their livelihood from agriculture (≥ 1 km^2 per farmer; Holmer, 2016; The Swedish Board of Agriculture, 2017).

GMSE models discrete individuals, which here include farmers, managers, and cranes. Each individual has potentially unique traits, which potentially affect their behavior. Each farmer owns a fixed number of contiguous landscape cells (L, see above) on which they can perform one or more types of actions. Each farmer has a budget B_f for performing actions, which can be broadly interpreted to encompass one or more factors limiting the total number of actions possible for the farmer during a single time step such as time, money, or available equipment. Thus, individual farmers' traits include their budget and farm location. Managers do not own landscape cells, but instead set policy that affects the costs of performing actions for farmers (C_{action}). Managers have traits in terms of a budget B_m for setting policy, which can be broadly interpreted to encompass the factors that limit the power of managers to make and enforce policy decisions. Managers either do not attempt to regulate crane population density, or attempt to maintain cranes at some target population density ($N^{\dagger}$) at every time step by setting policy (*i.e.*, costs for farmer actions). The traits for individual cranes are location, age and reproductive output. In this study, a single time step is modelled as one year. For definitions of state variable entities, landscape and model parameters, see Table S1.

3. Process overview and scheduling

The model includes four sub models operating in sequence over a single time step (Fig. 1). For detailed information on definition and use of parameters and modelling procedure, see '*Sub models*'.

The crane population sub model

The first sub model simulates the population dynamics of N_t cranes in time step *t*. Cranes arrive at a randomly selected cell within R_m cells of the one that they left in *t-1* with equal probability (crane cell location is randomly selected in t = 1), then feed R_{fe} times. Between each feeding, cranes move to a cell within R_m cells in any direction from the currently occupied cell on the landscape randomly selected with equal probability. Cranes then give birth to young, then potentially die of old age.

Initial population size for each simulation was independently sampled from the distribution of simulated terminal abundances from the 100 n_{rep} replicate simulations in years 1989-2019 (see '*Initializatio*n' & Fig. S1.). (Harris and Mirande, 2013)(Nilsson et al., 2018)

The observation sub model

The observation model uses a monitoring method in which cranes are counted with complete accuracy on a subset of the landscape (*i.e*, 10 × 10 cells) and density is then extrapolated to estimate the crane population size assuming the same density over the entire landscape. Hence, estimated population size $\hat{N}$ might deviate from the true *N*.

The manager sub model

The manager sub model assesses $\hat{N}$ in relation to a management target, and sets policy by defining action costs (C_{action}; these may be conceptualized as, *e.g.*, time, practical or monetary costs for different actions farmers may take to affect agricultural production, including culling or scaring cranes). Managers use an evolutionary algorithm to set costs for each action; this algorithm models the heuristic process of the manager considering different potential policies and choosing one that will result in a crane population density nearer to the manager's target (see 'Sub models'1). (Duthie et al., 2018)(Hamblin, 2013)(Duthie et al., 2018).Once the algorithm is completed, managers enact the cost of each available farmer action. The total budget for the manager was kept constant in the model (*i.e.*, B_m = 1000). See Section "*The evolutionary algorithm used in the manager and farmers sub models*" below for details on parameterization of the evolutionary algorithm.

The farmer sub model

In the farmer sub model, farmers implement actions with the objective to maximize their own agricultural production, constrained by the costs of individual actions as set by the manager's policy and the farmers' annual budgets (B_f). As with manager policy decisions, farmer decisions are chosen by running a single independent evolutionary algorithm for each farmer in each time step. Farmers recognize that the presence of cranes on a landscape cell has a negative effect on agricultural production, and will therefore adaptively use actions to try to effectively decrease the presence of cranes. Individual stakeholder decisions consequently affect cranes and agricultural production over multiple time steps. All actions of all farmers are performed in a random order so that, for example, one farmer does not do all of their scaring or culling before another farmer and thereby cause differences in farmer's agricultural production due to the order of farmer actions. Farmers can only take actions on land that they own.

4. Design concepts

Basic principles

To account for uncertainty at multiple points in a natural resource management process, the management strategy evaluation (*i.e.*, MSE) framework has been developed (Smith, 1999). MSE models have incorporated constant decision-making rules for single managers or farmers over time (Bunnefeld et al., 2013; Melbourne-Thomas et al., 2017; Milner-Gulland, 2011). However, the generalized management strategy evaluation framework (*i.e.*, GMSE) used in this study includes multiple independent stakeholders making individual decisions, as influenced by changes in resources, policy, and individual circumstance (Duthie et al., 2018). GMSE simulates the management strategy evaluation process in a way that models goal-oriented behavior and spatial distribution of individual stakeholders, managers and wildlife using an individual-based (*i.e.*, agent-based) framework (Bunnefeld et al., 2011; Duthie et al., 2018).

The observation model uses a "virtual ecologist" approach (Zurell et al., 2010) to model manager observation of the crane population.

Emergence
Crane population dynamics and agricultural production emerge over time, based on the individual decisions of the manager (*i.e.*, setting policy) and farmers (*i.e.*, taking action).

Adaptation
Evolutionary algorithms are useful heuristic tools that mimic the process of biological evolution to find solutions to highly complex problems (Hamblin, 2013). In our simulations, these complex problems refer to modelling the goal-oriented decision making of managers and farmers. Managers must attempt to use their available budget to set costs that keep crane populations near a pre-specified target, while farmers must attempt to use their available budget and any actions available to maximize agricultural production. The possible actions that farmers can take to reduce impact on agricultural production include non-lethal scaring, one action of which causes one crane to randomly relocate to a new cell before damaging crops (note, this could potentially result in the crane resettling on another cell owned by the acting farmer), or culling, one action of which causes one crane to be completely removed from the landscape before damaging crops (see scenario *a*-*b*). Decisions within the model are made under uncertainty, as is realistic for stakeholders in empirical social-ecological systems, and behavior is consequently heuristic often suboptimal. The Section "The evolutionary algorithm used in the manager and farmers sub models" below explains how the evolutionary algorithm is used to model manager and farmer decision making, with further detail of this general approach available in Supporting Information 1 of Duthie et al. (2018).

Objectives
The objective of the farmers is to maximize total agricultural production Y_f across all of the landscape cells that they own. Whereas the managers' objective is to set a policy (in terms of action costs) that minimizes the distance between the crane population target and the current population level (Duthie et al., 2018).

Learning
We included a learning process in each run of the evolutionary algorithm by seeding it with 20 copies of the strategy from the previous time step, allowing it to potentially learn from the previous time step and more efficiently find a successful strategy given similar conditions.

Prediction
Managers predict farmers´ actions in the evolutionary algorithm fitness function by assuming that the total number of actions in a time step will be proportional to those in the previous time step weighted by its cost, *e.g.*, if the cost doubles, then half the number of actions are predicted.

Sensing
Sensing is incorporated in the modelling in that farmers know that cranes decrease agricultural production. Farmers also know which landscape cells and what proportion of the landscape that they own, the costs of actions, and the probability of cranes landing back in their owned cells after being scared.

Managers estimate how many cranes are on the landscape. They also know the total number of actions taken by farmers in the previous time step, that culling decreases the number of cranes by one plus the number of offspring that it is expected to produce, and that scaring does not decrease the crane number.

Interaction

Cranes interact with the landscape by decreasing the agricultural production on the cell that they occupy and feed. Individual farmers, manager and cranes also interact with each other by adaptively making decisions based on current conditions. Managers make decisions based on estimated crane abundance and previous farmer actions, whereas farmers make decisions based on crane distribution on their land, management policy (*i.e.*, available actions) and budget. For details about interactions, see descriptions of sub models and sensing.

Stochasticity

In dynamic socio-ecological systems like the one studied here, it may be likely that empowerment of farmers (*i.e.*, budgets B_f) to perform actions varies among individuals at a given time step. To assess the effect of such potential variability in farmer's budgets, we repeated simulations for each of the scenarios with budgets varying among individuals by $B_f \pm 50$ in each time step. Stochasticity was also included in the crane population sub model by allowing movement of cranes to random cells within a given spatial range and by sampling the crane offspring number from a Poisson distribution with a rate parameter of $R_b = 0.118$.

Collectives

No collectives are implemented in the model.

Observation

We simulated increasingly empowered farmers, running simulation for a range of available budgets $B_f = \{50, 100, \ldots, 3950, 4000\}$, with 40 simulation replicates for each budget value across four management scenarios (*a*-*d*) over 30 years. We extracted selected performance metrics for each time step *t* (year 2020-2049); crane population size *N*, percentage of maximum agricultural production (*i.e.*, observed agricultural production (Y_f)/maximum agricultural production) over the individual farmers' cells (scenario *a*-*b)*, number of culled individuals per farmer (scenario *b*-*d*) and number of scaring actions performed per farmer (scenario *b & d*; Table 1).

5. Initialization

To assess the reproductive output for cranes, we used empirical time series data (1989-2019) of maximum number of cranes staging at a typical autumn staging site (*i.e.*, Lake Hornborga) in Sweden (Nilsson, 2016; Station and Västra, 2020) to replicate the exponential population growth and thus the reproductive output until year 2019 through simulations ($n_{rep} = 100$). The starting population in these simulations was based on the empirical data of ca 3000 cranes from 1989 (https://transtat.lansstyrelsen.se/, 2020; Nilsson, 2016), and we used these data to find a birth rate that produced a reasonable approximation to the empirical curve (see Fig. S1 in Supplementary material 2). The terminal abundances from the 100 n_{rep} were thereafter used as initial population sizes in simulations and expected offspring produced per adult individual (R_b) to simulate future population development from 2020 in the population sub model (see below).

From this initial population size, we simulated a further 30 years (2020-2049). Although it is inevitable that the European population of common cranes will eventually approach some carrying capacity *K*, there currently are no data to estimate *K* (Harris and Mirande, 2013); current exponential growth suggests that *K* is likely to be several degrees of magnitude larger than current population levels. We therefore allowed for unlimited exponential growth over the projected period. We model each time step *t* as a single year of migratory cranes arriving, using an agricultural landscape during staging, and departing at the end of the staging season (Nilsson et al., 2018).

6. Input data

The expected number of offspring per individual based on the simulation output was $R_b = 0.118$. When used to parametrize the model, it was sampled from a Poisson distribution with a rate parameter equal to R_b for each crane of age of $R_{ar} > 3$ (Nesbitt, 1992). Crane death occurred only as a consequence of culling or age $R_{ag} = 20$ (unpubl. ringing data). Cranes are known to repeatedly return to and move within a smaller part of staging sites (Nilsson et al., 2018), and each crane is thus modelled to occupy a landscape cell within $R_m = 4$ cells in any direction of the cell last occupied in the previous time step *t* (*see The population model*). Data to parameterize the individual effect of a crane on agricultural production are not known, but to model substantial but not complete loss of farmer's agricultural production (Montràs-Janer et al., 2019), we used 2 percent loss of remaining agricultural production Y_f when an individual crane moves to and feeds in a cell L_{xy}. Individual cranes move to and feed on 10 cells of the landscape during one time step, hence potentially causing the maximum equivalent of a 20% loss of agricultural production across landscape cells. If multiple cranes arrive at the same cell in a time step (or a single crane arrives at the same cell multiple times), then we assume that $(1\text{-}0.2)^{Nij}$ of the agricultural production is consumed on the cell.

7. Sub models

The model includes four sub -models operating in sequence over a single time step (Fig. 1). For detailed input data in each respective sub model, see Table S1.

The crane population sub model

The first sub model simulates the population dynamics of N_t cranes in time step *t*. At the beginning of the sub model, cranes arrive at a randomly selected cell within R_m cells of the one that they left in *t-1* with equal probability (if *t = 1*, then cranes are initialized in a random cell location), then feed R_{fe} times. Each feeding reduces current agricultural production on the occupied cell by 2%. Between each feeding, cranes move to a cell within R_m cells in any direction from the currently occupied cell on the landscape randomly selected with equal probability. After R_{fe} feeds have been completed, cranes give birth to offspring. The number of offspring that a crane produces is randomly sampled from a Poisson distribution with a rate parameter of $R_b = 0.118$. All offspring are placed on the landscape cell that their parent currently occupies, and neither parents nor offspring move during the remainder of the sub model. Parents that have exceeded the maximum age of $R_{ag} = 20$ die and are removed from the landscape. This concludes the population sub model.

The observation sub model

The observation lsub model uses a monitoring method in which cranes are counted with complete accuracy on a 10 × 10 cell square region of the landscape. The density of cranes on this 10 × 10 cell region is then calculated. The observation model assumes that the density of the entire 100 × 100 cell landscape equals the density of the 10 × 10 cell sampled region. The density is then extrapolated to estimate the crane population size assuming the same density over the entire landscape (*e.g.*, if 50 cranes were observed in the 10 × 10 cell sampled region, then the estimated density is 0.5 cranes per cell, so total population size is estimated as 100 × 100 × 0.5 = 500). Hence, estimated population size $\hat{N}$ might deviate from the true N due to sampling error in the 10 × 10 cell sampled region. With $\hat{N}$ calculated, this concludes the observation sub model.

The manager sub model

The manager sub model assesses $\hat{N}$ in relation to a management target and sets policy by defining action costs (C_{action}; these may be conceptualized as, *e.g.*, time, practical or monetary costs for different actions farmers may take to affect agricultural production, including culling or scaring cranes). Managers use an evolutionary algorithm to set costs for each action; this algorithm models the heuristic process of the manager considering different potential policies and choosing one that will result in a crane population density nearer to the manager's target see below; for complete documentation, see GMSE R package vignette 'The genetic algorithm': https://cran.r-project.org/package=GMSE) . To do this, managers make decisions based on current estimated crane abundance $\hat{N}$ and farmer actions from the previous time step (on the first time step, no knowledge of farmer actions is available, so costs are set randomly). Managers have a fixed budget of $B_m = 1000$, which can be used to set culling policy. The baseline cost of culling one crane is 10 for each farmer, but the manager can increase this cost by 1 unit for every 10 units that they spend from their own budget. Hence, if managers use the entirety of their total budget, then they can increase the cost that farmers must pay to cull a single crane up to 110, thereby resulting in fewer culled cranes because farmer budgets remain fixed within a given simulation.

The predicted effects of different farmer action costs are determined in the fitness function of the evolutionary algorithm and are calculated from the predicted effect that each action will have on crane abundance and the total number of actions predicted by the manager, as based on farmer actions in the previous time step (see Duthie et al., 2018 for details). For example, to predict the total number of culls that farmers will make in the current time step, managers use the calculation below

predicted_culls = culls_in_previous_time_step * (old_cost / new_cost).

The effect that a single cull is projected to have on population change is set as $0.9(1 + R_b)$; the 0.9 is a sensitivity parameter set to the GMSE default, and has been found to perform well for managers in decision-making (see Duthie et al. 2018 Supporting Information 1, and GMSE vignette 'The genetic algorithm': https://cran.r-project.org/package=GMSE). The value of predicted_culls times $0.9(1 + R_b)$ thereby defines the predicted effect of costs in the change in crane population as caused by culling. Scaring has no predicted effect on crane abundance, so managers do not directly modulate the cost of scaring to increase or decrease the crane abundance. Cost combinations that are predicted to result in crane population densities closer to the manager's target have higher fitness and are therefore more likely to be chosen as the final cost values set by the manager when the evolutionary algorithm terminates. Once the algorithm is completed, managers enact the cost of each available farmer action. The total budget for the manager was kept constant in the model (*i.e.*, $B_m = 1000$). See the section "*The evolutionary algorithm used in the manager and farmers sub models*" below for details on parameterization of the evolutionary algorithm.

The farmer sub model

In the farmer sub model, farmers implement actions with the objective to maximize their own agricultural production, constrained by the costs of individual actions as set by the manager's policy and the farmers' annual budgets (B_f). As with manager policy decisions, farmer decisions are chosen by running a single independent evolutionary algorithm for each farmer in each time step. Farmers recognize that the presence of cranes on a landscape cell has a negative effect on agricultural production, and farmers will therefore apply their annual budget to the available actions that most effectively decrease the presence of cranes. Fifty independent farmer decisions consequently affect cranes and agricultural production over multiple time steps. All actions of all farmers are performed in a random order so that, for example, one farmer does not do all of their scaring or culling before another farmer and thereby cause differences in farmer's agricultural production due to the order of farmer actions. Farmers can only take actions on land that they own.

Each farmer has a fixed budget of B_f, which they can potentially use to perform actions (scaring or culling) on their land. Each action has a baseline cost of 10, but this cost might be increased up to a maximum of 110 by the manager (see *The manager sub model* above). Farmers perceive each culling action to decrease the number of cranes on their land by $(1 + R_b)$ and each scaring action to decrease the number of cranes on their land by (1 – 1/50). The decrement of 1/50 accounts for the probability that when a scared crane moves, it will land back on a landscape cell owned by the scaring farmer. All else being equal, farmers therefore recognize that culling is more effective at reducing cranes on their land than scaring, but the total effectiveness of any combination of scaring and culling will also be modulated by the cost of actions set by the manager. Combinations of actions that are found by the evolutionary algorithm to minimize the number of cranes on a focal farmer's landscape will have a higher fitness and therefore be more likely to be selected as the focal farmer's final set of actions when the evolutionary algorithm terminates. Once the evolutionary algorithm has finished, actions of all farmers are performed in a random order.

The evolutionary algorithm used in the manager and farmers sub models

This section provides more detail on the evolutionary algorithm described in the sections "*The manager sub model*" and "*The farmer sub model*" above. For complete documentation, see GMSE R package vignette 'The genetic algorithm': https://cran.r-project.org/package=GMSE) as well as Duthie et al (2018). Code underlying the evolutionary algorithm of GMSE v0.6.0.4 is available on GitHub: https://github.com/ConFooBio/gmse/blob/1c295dd53a63cc9aa905f6934cf2800711212a79/src/game.c All GMSE code is publicly available in the GMSE GitHub repository: https://github.com/ConFooBio/gmse

GMSE uses an evolutionary algorithm (also described as a 'genetic algorithm') to model agent decision-making (https://confoobio.github.io/gmse/articles/SI1.html). Evolutionary algorithms are useful heuristic tools that mimic the process of biological evolution to find solutions to highly complex problems (Hamblin 2013). In our simulations, these complex problems refer to modelling the adaptive and goal-oriented decision making of managers and farmers. Managers must attempt to use their available budget to set costs that keep crane populations near a pre-specified target, while farmers must attempt to use their available budget and any actions available to maximize agricultural production. Decisions within the model are made under uncertainty, as is realistic for stakeholders in empirical social-ecological systems, and behavior is consequently heuristic often suboptimal.

Evolutionary algorithms are inspired by the process of evolution by natural selection. For all agents in GMSE, this process includes the initialization of a population of strategies (*e.g.*, manager cost combinations or farmer actions), strategy crossover (recombination), mutation, cost constraint (to avoid going over budget), fitness evaluation, selection, and replacement of original strategies (see Figure S1 below, which also appears in GMSE documentation and Duthie et al. 2018 Supporting Information 1).

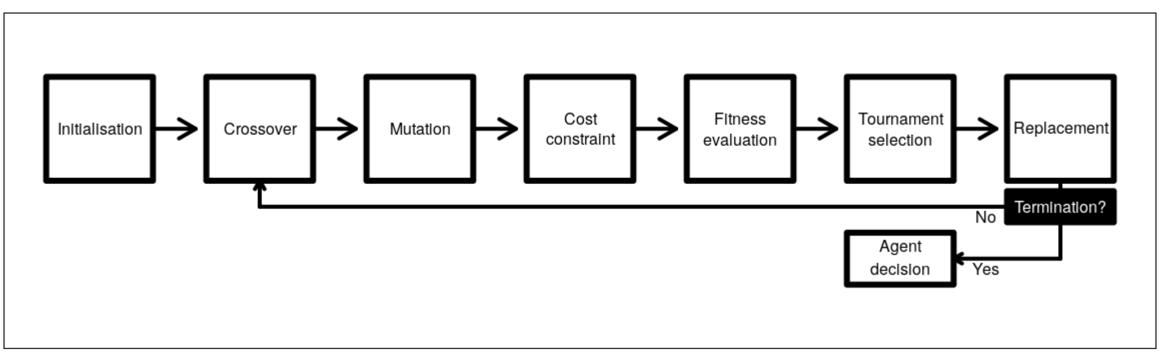

Figure S1: Evolutionary algorithm of GMSE R package

To implement the process described above (Figure S1), the evolutionary algorithm of GMSE is parameterized using up to nine arguments to the gmse function used for simulation, including ga_popsize, ga_mingen, ga_seedrep, ga_sampleK, ga_chooseK, ga_mutation, ga_crossover, ga_converge_crit, and group_think. Details on each parameter are provided in the GMSE documentation and GMSE vignettes.

Individual steps of the evolutionary algorithm include the following for agents (managers or farmers):

- **Initialization**: Every time that an agent makes a decision (i.e., each time step), ga_popsize temporary agent decisions are created, each of which includes a set of decisions that could potentially be made by the agent (e.g., the number of scares or culls for a farmer). Of these ga_popsize agent decisions, ga_seedrep are replicates of the decisions made in the previous time step of the simulation by the agent (unless the simulation is in its first time step, in which case all ga_popsize are initialised to be random decisions). The remaining ga_popsize - ga_seedrep are randomly generated sets of decisions.
- **Crossover:** Actions in the ga_popsize strategies from the population are randomly swapped with some fixed probability of ga_crossover.
- **Mutation:** Actions in strategies from the population randomly mutate with a probability of ga_mutation; mutation causes an increment or decrement of a particular action by one.
- **Cost constraint:** If the budget of any strategy in the population (i.e., B_m or B_f) has been exceeded, then actions are removed randomly until the strategy is within its budget.
- **Fitness evaluation**: The fitness of each strategy in the population is calculated (see *"The manager sub model"* and *"The farmer sub model"* above for fitness explanations).
- **Tournament selection:** Strategies within the population are randomly sampled, and a subset of the sampled strategies with the highest fitness are selected to be retained. A random subset of ga_sampleK strategies is sampled with replacement, and the ga_chooseK highest fitness strategies are chosen and retained; tournament selection continues in this fashion until ga_popsize new decision sets are retained.
- **Replacement:** The highest fitness strategies from the tournament selection replace the strategies in the population, and thereby become the new population of strategies.
- **Termination**: If termination conditions are not met, then the new population of strategies undergoes a new round of crossover, mutation, cost constraint, fitness evaluation, and tournament selection. If termination conditions are met, then the highest fitness strategy in the population is selected as the agent's new strategy (if there is a tie between two or more highest fitness strategies, then one of those strategies is randomly selected with equal probability). Termination conditions are not met unless either the percent increase in highest fitness decision set is greater than ga_converg_crit, or if fewer than ga_mingen iterations of crossover, mutation, and selection have passed.

In a single time step, it is possible to set group_think = TRUE, in which case every farmer will make the same set of decisions (requiring only one call to the evolutionary algorithm for all farmers). For all simulations except those of scenario *a*, we used the default group_think = FALSE. Since farmers cannot take any actions in scenario *a*, we set group_think = TRUE to lower simulation time. Remaining parameters were set as below.

GMSE argument	**Value**	**Comment**

ga_popsize	100	Set to the GMSE default value
ga_mingen	200	Increased from GMSE default 40 to improve decision-making precision
ga_seedrep	20	Set to the GMSE default value
ga_sampleK	20	Set to the GMSE default value
ga_chooseK	2	Set to the GMSE default value
ga_mutation	0.1	Set to the GMSE default value
ga_crossover	0.1	Set to the GMSE default value
ga_converge_crit	0.01	Decreased from GMSE default 0.1 to improve decision-making precision

GMSE defaults were mostly used, as these have performed well in exploratory simulations and GMSE package development (Duthie et al. 2018). Where deviations from default values were chosen, it was to increase the ability of the evolutionary algorithm to find adaptive decision-making solutions.

Code for replicating simulations of sub models

To replicate the simulations of this manuscript, it is necessary to install the GMSE package version 0.6.0.4. As of the time of writing, this is the most recent version of GMSE available on CRAN, and can be downloaded in R as below.

```
install.packages(GMSE);
```

GMSE v0.6.0.4 can also be installed directly from GitHub using the devtools packages (Wickham et al., 2020).

```
devtools::install_github("ConFooBio/GMSE", ref = "v0.6.0.4");
```

All code underlying GMSE is open source and publicly available on the Comprehensive R Archive Repository (https://cran.r-project.org/package=GMSE) and on GitHub (https://github.com/ConFooBio/gmse)

Results from the gmse function are summarized with a custom written function summarise_cranes, which is described below.

```
# Custom function to get crane results
summarise_cranes <- function(sim){
  mat <- matrix(data = 0, nrow = length(sim$resource), ncol = 9);
  colnames(mat) <- c("generation", "user_budget", "pop_size", "yield_mn",
                     "yield_sd", "scare_c", "cull_c", "scares", "culls");
  for(i in 1:length(sim$resource)){
    mat[i, 1] <- i;
    mat[i, 2] <- sim$paras[i, 98];
    mat[i, 3] <- dim(sim$resource[[i]])[1];
    mat[i, 4] <- mean(sim$agents[[i]][sim$agents[[i]][,2] > 0, 16]);
    mat[i, 5] <- sd(sim$agents[[i]][sim$agents[[i]][,2] > 0, 16]);
    mat[i, 6] <- sim$action[[i]][3, 8, 1];
    mat[i, 7] <- sim$action[[i]][3, 9, 1];
    mat[i, 8] <- sum(sim$action[[i]][1, 8, ]);
    mat[i, 9] <- sum(sim$action[[i]][1, 9, ]);
  }
```

```
    return(mat);
}
```

Simulations were run using custom written functions scenario_a, scenario_b, scenario_c, and scenario_d. Each function took a vector of user budgets $\{50, 100, \ldots, 3950, 4000\}$ as an argument, which was set as the variable budgets.

```
budgets <- seq(from = 50, to = 4000, by = 50);
```

A file named abfile also specified the location of initialized crane abundances, as explained in the main text.

```
abfile  <- "starting_values.csv";
```

The contents of abfile are the 100 starting abundances for simulations, which are printed below.

```
print(abundances[,2]);
##   [1] 18111 18167 18265 17873 17768 19138 18332 17002 18475 17500 18040 18106
##  [13] 19182 18011 17524 17618 18310 18281 17867 18270 18030 18264 17925 17703
##  [25] 17870 18589 18512 18868 18532 18081 17133 17176 17984 17760 19020 18581
##  [37] 18872 17777 18233 18648 17130 17169 17830 18206 18205 18657 17695 17139
##  [49] 18646 18321 17088 18590 17584 18733 17557 18290 19680 18702 18054 17120
##  [61] 17173 17015 17500 18818 18337 18316 18181 17977 17536 17809 17197 17297
##  [73] 18734 18515 17280 19607 18723 17292 18948 18527 18126 18008 16908 18001
##  [85] 17338 17589 17607 17631 18022 18682 18362 16857 17390 17823 18264 18309
##  [97] 18403 18720 16881 17942
```

A single replicate of simulation of scenario *a* is run using the code below.

```
# Scenario a function
scenario_a <- function(user_budgets, filename, abund_file){
    starting_table <- read.csv(file = abund_file);
    for(budget in user_budgets){
        ini_abundance  <- sample(x = starting_table[,2], size = 1);
        sim <- gmse(manager_budget = 1000, user_budget = budget,
                    manage_target = 100000,  RESOURCE_ini = ini_abundance,
                    plotting = FALSE,  stakeholders = 50, times_feeding = 10,
                    land_ownership = TRUE, public_land = 0.0, remove_pr = 0,
                    scaring = FALSE, culling = FALSE, lambda = 0.118,
                    time_max = 30, res_death_type = 1, res_movement = 4,
                    res_consume = 0.02, res_birth_K = 200000, observe_type = 0,
                    agent_view = 10, max_ages = 20, usr_budget_rng = 0,
                    converge_crit = 0.01, ga_mingen = 200, age_repr = 4,
                    group_think = TRUE); # Since no actions, group_think okay.
        s_mat <- summarise_cranes(sim);
        write.table(s_mat, filename, sep = ",", row.names = FALSE,
                    col.names = !file.exists(filename), append = TRUE);
        rm(sim);
        rm(s_mat);
        gc();
    }
    return("Done!");
}
# Run scenario a
scenario_a(user_budgets = budgets, filename = "out.csv", abund_file = abfile);
```

A single replicate of simulation of scenario 2 is run using the code below.

```r
# Scenario b function
scenario_b <- function(user_budgets, filename, abund_file){
   starting_table <- read.csv(file = abund_file);
   for(budget in user_budgets){
      ini_abundance  <- sample(x = starting_table[,2], size = 1);
      sim <- gmse(manager_budget = 1000, user_budget = budget,
               manage_target = 100000,  RESOURCE_ini = ini_abundance,
               plotting = FALSE,  stakeholders = 50, times_feeding = 10,
               land_ownership = TRUE, public_land = 0.0, remove_pr = 0,
               scaring = TRUE, culling = TRUE, lambda = 0.118,
               time_max = 30, res_death_type = 1, res_movement = 4,
               res_consume = 0.02, res_birth_K = 200000, observe_type = 0,
               agent_view = 10, max_ages = 20, usr_budget_rng = 0,
               converge_crit = 0.01, ga_mingen = 200, age_repr = 4,
               group_think = FALSE);
      s_mat <- summarise_cranes(sim);
      write.table(s_mat, filename, sep = ",", row.names = FALSE,
               col.names = !file.exists(filename), append = TRUE);
      rm(sim);
      rm(s_mat);
      gc();
   }
   return("Done!");
}
# Run scenario b
scenario_b(user_budgets = budgets, filename = "out.csv", abund_file = abfile);
```

A single replicate of simulation of scenario 3 is run using the code below.

```r
# Scenario c function
scenario_c <- function(user_budgets, filename, abund_file){
   starting_table <- read.csv(file = abund_file);
   for(budget in user_budgets){
      ini_abundance  <- sample(x = starting_table[,2], size = 1);
      sim <- gmse(manager_budget = 1000, user_budget = budget,
               manage_target = 100000,  RESOURCE_ini = ini_abundance,
               plotting = FALSE, stakeholders = 50, times_feeding = 10,
               land_ownership = TRUE, public_land = 0.0, remove_pr = 0,
               scaring = FALSE, culling = TRUE, lambda = 0.118,
               time_max = 30, res_death_type = 1, res_movement = 4,
               res_consume = 0.02, res_birth_K = 200000, observe_type = 0,
               agent_view = 10, max_ages = 20, usr_budget_rng = 0,
               converge_crit = 0.01, ga_mingen = 200, age_repr = 4,
               group_think = FALSE);
      s_mat <- summarise_cranes(sim);
      write.table(s_mat, filename, sep = ",", row.names = FALSE,
               col.names = !file.exists(filename), append = TRUE);
      rm(sim);
      rm(s_mat);
      gc();
   }
   return("Done!");
```

```
}
# Run scenario c
scenario_c(user_budgets = budgets, filename = "out.csv", abund_file = abfile);
```

A single replicate of simulation of scenario 4 is run using the code below.

```
# Scenario d function
scenario_d <- function(user_budgets, filename, abund_file){
  starting_table <- read.csv(file = abund_file);
  for(budget in user_budgets){
    ini_abundance  <- sample(x = starting_table[,2], size = 1);
    sim <- gmse(manager_budget = 1000, user_budget = budget,
                manage_target = 15000,  RESOURCE_ini = ini_abundance,
                plotting = FALSE, stakeholders = 50, times_feeding = 10,
                land_ownership = TRUE, public_land = 0.0, remove_pr = 0,
                scaring = TRUE, culling = TRUE, lambda = 0.118,
                time_max = 30, res_death_type = 1, res_movement = 4,
                res_consume = 0.02, res_birth_K = 200000, observe_type = 0,
                agent_view = 10, max_ages = 20, usr_budget_rng = 0,
                converge_crit = 0.01, ga_mingen = 200, age_repr = 4,
                group_think = FALSE);
    s_mat <- summarise_cranes(sim);
    write.table(s_mat, filename, sep = ",", row.names = FALSE,
                col.names = !file.exists(filename), append = TRUE);
    rm(sim);
    rm(s_mat);
    gc();
  }
  return("Done!");
}
# Run scenario d
scenario_d(user_budgets = budgets, filename = "out.csv", abund_file = abfile);
```

To vary user budgets, scenarios 1-4 can be run with usr_budget_rng = 50 instead of usr_budget_rng = 0.

Table S1. Definition of state variable entities, landscape and model parameters used in the modelling.

Variable	Description	Sub model	Parameter value	Unit	Reference
Entities					
Y_f	total agricultural production (*i.e*, yield) over landscape cells owned by an individual farmer				
t	time step				
$(1-p)^{Nij}$	percentage loss of total agri. prod. at t when multiple cranes occupies a landscape cell L_{xy}				
N_t	crane population at time step t				
$\hat{N}$	estimated size of crane population through manager observation				
Land-scape					
L	the agricultural landscape		100x100	cells	
L_{xy}	landscape cell at a given location (x-y) at the agricultural landscape				
Model					
B_f	individual farmer budgets (*i.e.*, time, practical monetary resources) to perform actions	farmer	50-4000	no unit	
C_{action}	cost for farmers to perform actions, set by the manager through policy	farmer	based on policy	no unit	
B_m	manager budget to set policy	manager	1000	no unit	
$N^{\dagger}$	target population density set by the manager through policy	manager	see Table 1	crane ind.	
R_{ar}	age of first reproduction for cranes	crane	>3	years	Nesbitt 1992
R_{ag}	maximum age of a crane	crane	20	years	unpubl. ringing data
R_b	mean number of offspring per crane in each time step	crane	0.118	offspring per crane	
R_m	potential movement range of a crane (landscape cells in any direction) from previous time step $t-1$	crane	4	cells	Nilsson *et al.* 2018
R_{fe}	times feeding by an individual on the landscape in a time step t	crane	10	cells	
N_{t+0}	crane population at initial time step	crane	16857-19680, mean:18034	crane ind.	

References

Bunnefeld, N., Edwards, C.T.T., Atickem, A., Hailu, F., Milner-Gulland, E.J., 2013. Incentivizing Monitoring and Compliance in Trophy Hunting. Conserv. Biol. 27, 1344–1354. https://doi.org/10.1111/cobi.12120

Bunnefeld, N., Hoshino, E., Milner-Gulland, E.J., 2011. Management strategy evaluation: a powerful tool for conservation? Trends Ecol. Evol. 26, 441–447. https://doi.org/10.1016/j.tree.2011.05.003

Duthie, A.B., Cusack, J.J., Jones, I.L., Minderman, J., Nilsen, E.B., Pozo, R.A., Rakotonarivo, O.S., Van Moorter, B., Bunnefeld, N., 2018. GMSE: An r package for generalised management strategy evaluation. Methods Ecol. Evol. 9, 2396–2401. https://doi.org/10.1111/2041-210X.13091

EEA, n.d. HaBiDeS [WWW Document]. URL https://www.eionet.europa.eu/software/habides (accessed 5.27.19).

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe'er, G., Piou, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L., 2006. A standard protocol for describing individual-based and agent-based models. Ecol. Modell. 198, 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

Grimm, V., Railsback, S.F., Vincenot, C.E., Berger, U., Gallagher, C., Deangelis, D.L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., Johnston, A.S.A., Milles, A., Nabe-Nielsen, J., Polhill, J.G., Radchuk, V., Rohwäder, M.S., Stillman, R.A., Thiele, J.C., Ayllón, D., 2020. The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. JASSS 23. https://doi.org/10.18564/jasss.4259

Hamblin, S., 2013. On the practical usage of genetic algorithms in ecology and evolution. Methods Ecol. Evol. https://doi.org/10.1111/2041-210X.12000

Harris, J., Mirande, C., 2013. A global overview of cranes: status, threats and conservation priorities. Chinese Birds 4, 189–209. https://doi.org/10.5122/cbirds.2013.0025

Holmer, J., 2016. Statistiskt meddelande - Heltidsjordbruket i Sverige 2016.

https://transtat.lansstyrelsen.se/ [WWW Document], 2020. URL https://transtat.lansstyrelsen.se/ (accessed 12.18.20).

Melbourne-Thomas, J., Constable, A.J., Fulton, E.A., Corney, S.P., Trebilco, R., Hobday, A.J., Blanchard, J.L., Boschetti, F., Bustamante, R.H., Cropp, R., Everett, J.D., Fleming, A., Galton-Fenzi, B., Goldsworthy, S.D., Lenton, A., Lara-Lopez, A., Little, R., Marzloff, M.P., Matear, R., Mongin, M., Plaganyi, E., Proctor, R., Risbey, J.S., Robson, B.J., Smith, D.C., Sumner, M.D., Putten, E.I. van, 2017. Integrated modelling to support decision-making for marine social-ecological systems in Australia. ICES J. Mar. Sci. 74, 2298–2308. https://doi.org/10.1093/ICESJMS/FSX078

Milner-Gulland, E.J., 2011. Integrating fisheries approaches and household utility models for improved resource management. Proc. Natl. Acad. Sci. U. S. A. 108, 1741–1746. https://doi.org/10.1073/pnas.1010533108

Monträs-Janer, T., Knape, J., Nilsson, L., Tombre, I., Pärt, T., Månsson, J., 2019. Relating national levels of crop damage to the abundance of large grazing birds: Implications for management. J. Appl. Ecol. 56, 2286–2297. https://doi.org/10.1111/1365-2664.13457

Nesbitt, S.A., 1992. First Reproductive Success and Individual Productivity in Sandhill Cranes. J. Wildl.

Manage. 56, 573. https://doi.org/10.2307/3808874

Nilsson, L., 2016. Common cranes in agricultural landscapes- linking space use and foraging patterns to conservation and damage prevention. Swedish University of Agricultural Sciences.

Nilsson, L., Aronsson, M., Persson, J., Månsson, J., 2018. Drifting space use of common cranes—Is there a mismatch between daytime behaviour and management? Ecol. Indic. 85, 556–562. https://doi.org/10.1016/j.ecolind.2017.11.007

Smith, A., 1999. Implementing effective fisheries-management systems – management strategy evaluation and the Australian partnership approach. ICES J. Mar. Sci. 56, 967–979. https://doi.org/10.1006/jmsc.1999.0540

Hornborgasjöns Fältstation, Länsstyrelsen Västra Götaland., 2020.[WWW Document]. URL https://transtat.lansstyrelsen.se/ (accessed 5.12.20).

The Swedish Board of Agriculture, 2017. SAM14 [WWW Document]. http://www.jordbruksverket.se/amnesomraden/stod/jordbrukarstod.4.4b3f0532150f4b827c7e3735.html.

Wickham, H., J. Hester, Chang., W., 2020. Tools to Make Developing R Packages Easier • devtools [WWW Document]. URL https://devtools.r-lib.org/ (accessed 12.18.20).

Zurell, Damaris, Berger, U., Cabral, J.S., Jeltsch, Florian, Meynard, C.N., Münkemüller, T., Nehrbass, N., Pagel, Jörn, Reineking, B., Schröder, Boris, Grimm, V., Zurell, D, Schröder, B, Cabral, -J S, Jeltsch, F, Pagel, J, 2010. The virtual ecologist approach: simulating data and observers. Oikos 119, 622–635. https://doi.org/10.1111/j.1600-0706.2009.18284.x