

Article

Utility of Particle Swarm Optimization in Statistical Population Reconstruction

Sergey S. Berg 

Department of Computer and Information Sciences, University of St. Thomas, 2115 Summit Avenue, St. Paul, MN 55105, USA; berg@stthomas.edu; Tel.: +1-651-962-5382

Abstract: Statistical population reconstruction models based on maximum likelihood and minimum chi-square objective functions provide a robust and versatile approach to estimating the demographic dynamics of harvested populations of wildlife. These models employ numerical optimization techniques to determine which set of model parameters best describes observed age-at-harvest, catch-effort, and other auxiliary field data. Although numerous optimization methods have been used in the past, the benefits of using particle swarm optimization (PSO) have yet to be explored. Using a harvested population of North American river otter (*Lontra canadensis*) in Indiana as a case study, we investigated the performance of population reconstruction using particle swarm optimization, spectral projected gradient (SPG), Nelder–Mead, and Broyden–Fletcher–Goldfarb–Shanno (BFGS) methods. We used Monte Carlo studies to simulate populations under a wide range of conditions to compare the relative performance of population reconstruction models using each of the four optimization methods. We found that using particle swarm optimization consistently and significantly improved model stability and precision when compared with other numerical optimization methods that may be used in statistical population reconstruction. Given that these models are frequently used to guide management decisions and set harvest limits, we encourage management agencies to adopt this more precise method of estimating model parameters and corresponding population abundance. These results illustrate the benefits of using particle swarm optimization, caution against relying on the results of population reconstruction based on optimization methods that are highly dependent on initial conditions, and reinforce the need to ensure model convergence to a global rather than a local maximum.



Citation: Berg, S.S. Utility of Particle Swarm Optimization in Statistical Population Reconstruction.

Mathematics **2023**, *11*, 827. <https://doi.org/10.3390/math11040827>

Academic Editor: Jinfeng Xu

Received: 29 December 2022

Revised: 28 January 2023

Accepted: 3 February 2023

Published: 6 February 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: population analysis; abundance estimation; age-at-harvest; harvest mortality; statistical population reconstruction; maximum likelihood estimation; numerical optimization; particle swarm optimization

MSC: 62P12; 65K10

1. Introduction

Statistical population reconstruction (SPR) using integrated population models (IPMs) is a robust and cost-effective means of estimating wildlife population abundance across an entire state or other large spatial scale at which harvest-related management typically occurs. By incorporating relevant information from multiple sources, such as age-at-harvest and catch-effort data, as well as auxiliary information from index surveys [1], mark–recapture analyses [2], and radio-telemetry studies [3] when available, these models provide a flexible framework to simultaneously estimate various demographic parameters (e.g., annual abundance, recruitment, and survival). Such models continue to see increased use as a valuable tool for monitoring wildlife population trends throughout the world, including moose (*Alces alces*) in Minnesota [4], grizzly bears (*Ursus arctos*) in British Columbia [5], and rock ptarmigans (*Lagopus muta*) in Iceland [6].

Population reconstruction typically begins by specifying a state–space model that links yearly estimates of population abundance with demographics rates, such as recruitment, natural (i.e., non-harvest) survival, and harvest vulnerability. An objective function is then defined to quantify the likelihood of observing the age-at-harvest data collected in the field given the demographic rates in the state–space model [7–10]. The most common approach to defining this objective function is to construct a product-multinomial likelihood for each cohort in the state–space model, using initial cohort abundance, natural survival, and harvest vulnerability as model parameters [1,11,12]. An alternative to this maximum likelihood (ML) formulation is instead to use a minimum chi-square (MCS) objective function to achieve the same goal [2,4,10]. Although MCS is used less frequently than ML, it does provide several advantages, including being more likely to converge to a global optimum and avoiding potential numerical problems associated with large combinatoric terms used in multinomial likelihoods [2]. Further, preliminary simulation studies suggest that both approaches yield similar levels of accuracy and precision under a range of conditions [10].

Regardless of whether an ML or an MCS model is employed, numerical optimization is then used to find either the maximum or minimum of the corresponding objective function in a multidimensional space to determine which set of model parameters best describes the observed age-at-harvest data, as well as any auxiliary data from radio-telemetry, mark–recapture, or index surveys. Several different optimization algorithms have been used to estimate these model parameters in the past, including a bounded form of the Nelder–Mead method [3] and a spectral projected gradient method [4,10]. Unfortunately, very little attention has been devoted to how these different optimization methods may influence the precision and stability of statistical population reconstruction. Complex nonlinear models, such as those used in population reconstruction, can be very difficult to numerically optimize, and problems of false maxima and unstable variance estimates are common to many numerical optimization methods [13]. This would suggest that the choice of numerical optimization method is an important consideration in statistical population reconstruction.

Although previous population reconstruction models have used only a few different numerical optimization methods to estimate model parameters, there is a wide range of other methods that should also be explored. Particle swarm optimization (PSO), for example, is a heuristic optimization method inspired by the social behavior and movement of animals in a flock of birds or a school of fish. Developed in 1995, this method employs a group of random particles meant to represent individual birds or fish, each of which searches the multidimensional parameter space of the objective function for a global maximum [14]. Although each particle is treated as an individual, their movements are guided not just by their own best-known position but also by the best-known position of other particles in the swarm [15–17]. This balance of global and local exploration allows particle swarm optimization to be used in nonlinear models that contain many parameters, and it does not require initial parameter guesses or derivatives of the objective function [18]. The use of this method has been explored in disciplines ranging from chemical engineering [18] to finance [19] and may provide a suitable alternative to the optimization methods currently employed in statistical population reconstruction.

The purpose of our paper is to provide wildlife managers and agencies with guidance on how the choice of a specific numerical optimization method may influence the accuracy of population reconstruction and, by extension, the efficacy of management decisions regarding season lengths and catch limits. To the best of our knowledge, this is the first comprehensive assessment of how the choice of numerical optimization methods can impact the results of reconstruction models. We used Monte Carlo simulations of populations with varying levels of abundance, natural survival, and harvest rates to compare the accuracy and precision of ML statistical population reconstruction results derived using different numerical optimization methods. We also investigate the effectiveness of PSO as a more efficient and accurate optimization method for identifying the best set of model parameters in population reconstruction. Finally, we illustrate how these results can be applied to

help guide management and population reconstruction of a harvested population of North American river otter (*Lontra canadensis*) in Indiana, USA. The main contributions of this paper can be summarized as follows:

- Discussing the role of numerical optimization methods in statistical population reconstruction models.
- Providing a quantitative comparison of how different numerical optimization methods impact the precision of reconstruction estimates.
- Recommending the use of particle swarm optimization as a more effective alternative to other numerical optimization methods that may be used in reconstruction models.
- Presenting a reconstruction model of river otter in Indiana as a case study of how using other numerical optimization methods may result in less accurate assessments of population abundance.

The remainder of this paper is organized as follows. The underlying model structure, numerical optimization, and demographic data used in the population reconstruction is detailed in Section 2. The results of the corresponding simulation and cases studies are presented in Section 3. Section 4 describes implications of the results to future and ongoing reconstructions efforts, as well as a detailed discussion of the population estimates produced for the case study of river otter in Indiana. Finally, Section 5 presents a conclusion and recommendations for wildlife management agencies.

2. Materials and Methods

2.1. Formulation of the Population Reconstruction Model

2.1.1. Available Demographic Data

From 2015 to 2020, the Indiana Department of Fish and Wildlife Resources (IN DNR) recorded all river otter harvested in Indiana during the regulated furbearer trapping seasons and aged harvested otter that were mandatorily submitted by trappers by using a combination of microscopic counts of cementum annuli and X-ray examination of pulp cavities. Because not all harvested river otter were successfully aged due to loss in transit and limited noncompliance by trappers (mean number aged annually = 98%, range = 94–100%), age-at-harvest numbers were inflated accordingly by assuming that the aged sample is representative of the overall harvest [2,3,10].

IN DNR also began sending annual surveys in 2017 to individuals who purchased a trapping license (mean response rate = 24%, range = 21–31%) to estimate the mean number of trap nights per licensed trapper, scaled to a mean of one. When survey results were unavailable, we assumed that the mean number of trap nights was equal to one (indicating average number of trap nights per licensed trapper when compared with other years) and used the combination of number of licenses sold and scaled mean number of trap nights per licensed trapper as an estimate of yearly catch effort (Table 1).

Table 1. Age-at-harvest numbers (and percentages) and estimated catch effort (in terms of total trap-nights, scaled to a mean of one) for North American river otter in Indiana, USA, 2015–2020.

Year	Young-of-the-Year	Yearlings	Adults	Total	Catch Effort
2015	242 (40.1%)	181 (30.0%)	180 (29.9%)	603	1.155
2016	158 (30.5%)	208 (40.2%)	152 (29.3%)	518	0.959
2017	232 (46.7%)	179 (36.0%)	86 (17.3%)	497	1.113
2018	142 (24.2%)	241 (41.1%)	204 (34.8%)	587	0.809
2019	127 (20.7%)	281 (45.8%)	206 (33.6%)	614	0.953
2020	174 (28.2%)	260 (42.2%)	182 (29.5%)	616	1.011

2.1.2. Underlying State–Space Model

The basic form of a reconstruction model begins with a deterministic population projection matrix [20,21] to describe the development of the underlying population sizes over time and to provide a means of projecting the expected harvest data to compare

with the observed harvest data [8,9,22]. Consider a hypothetical population of animals divided into three age classes (young-of-the-year, yearlings, and adults) harvested over Y consecutive years, where N_{ij} and h_{ij} are the abundance immediately before harvest and the number subsequently harvested, respectively, of animals of age-class j ($j = 1, 2, 3$) in year i ($i = 1, \dots, Y$).

Under this framework, all individuals born during the same year constitute a single cohort that is subsequently subjected to annual mortality from harvest and natural causes. Using an age-at-harvest matrix (Table 1), we can represent each cohort as a separate diagonal, where the observed counts, h_{ij} , are a function of the initial abundance of the corresponding cohort and annual rates of harvest mortality and natural survival (to be estimated as parameters).

2.1.3. Multinomial Likelihood Objective Function

Using the ML method to define the corresponding objective function, we model each of the diagonals in the age-at-harvest matrix (Table 1) as independent multinomial distributions. The likelihood for the diagonal represented by animals of age-class $j = 3$ (i.e., adults) already present in the population in year $i = 1$ (i.e., N_{13}), for example, can be written as follows:

$$l_{13} = \binom{N_{13}}{h_{13}} (P_{13})^{h_{13}} (1 - P_{13})^{N_{13} - h_{13}}, \tag{1}$$

where P_{13} is the probability of an adult being harvested in year 1. This likelihood is based on a binomial distribution of the number of animals in each cohort harvested and not harvested each year and forms the basis of most statistical population reconstruction models [1,11,23,24].

A common parameterization for the harvest probability P_{13} that is used in both population reconstruction [1,3,12,24] and other models of animal abundance [25,26] is as follows:

$$P_{13} = \left(1 - e^{-f_1 C_3} \right), \tag{2}$$

where f_1 is the catch effort in year 1 and C_3 represents the harvest vulnerability coefficient for animals of age-class $j = 3$. This parameterization provides a direct means of converting the observed catch effort (Table 1) into a harvest probability by estimating the vulnerability coefficient as a parameter in the model [3,10,23].

The full population reconstruction model requires development of similar likelihoods for each of the other cohorts or diagonals in the age-at-harvest matrix. The overall likelihood for the entire age-at-harvest matrix can then be written as a product of these individual cohort likelihoods:

$$L_{Age-at-harvest} = \prod_{i=1}^Y l_{i1} \times \prod_{j=2}^3 l_{1j}, \tag{3}$$

where l_{ij} represents the cohort-specific age-at-harvest likelihoods described above [11,12,23,24,27].

This age-at-harvest likelihood alone is over-parameterized and incapable of estimating parameters for initial cohort abundance and separate values for both natural survival and harvest vulnerability. At least one of these parameters must be estimated independently of the age-at-harvest likelihood with the use of some additional source of field information [1,11,24,28]. One of the most common sources of this additional information is catch per unit effort (CPUE), which can be used to model the total annual harvest as a binomially distributed function of catch effort:

$$L_{Catch-effort} = \prod_{i=1}^Y \binom{\sum_{j=1}^3 N_{ij}}{\sum_{j=1}^3 h_{ij}} (P_i)^{\sum_{j=1}^3 h_{ij}} (1 - P_i)^{\sum_{j=1}^3 (N_{ij} - h_{ij})}, \tag{4}$$

where P_i is the mean probability of beings harvested in year i . Much like the age-at-harvest likelihood, this likelihood is based on a binomial distribution of the total number of animals

harvested and not harvested each year [1,12,24,27]. This likelihood can also be expanded to use separate harvest probabilities for each of the different age-classes if needed [9,27].

Using both the age-at-harvest and the catch-effort likelihoods, we can write the full joint multinomial likelihood that is used in statistical population reconstruction models as follows:

$$L_{Joint} = L_{Age-at-harvest} \times L_{Catch-effort} \tag{5}$$

This joint likelihood can also be expanded to include additional information from index surveys [1,12], radio-telemetry studies [3,12,29], or other relevant data collected during the same time as the age-at-harvest and catch-effort data. However, because these other data are not always as readily available, the joint age-at-harvest and catch-effort likelihood serves as the foundational objective function in most statistical population reconstruction models [1,3,10,24,27].

2.1.4. Parameter Estimation

Once an appropriate objective function has been defined, numerical optimization can be used to find the global maximum of this likelihood function to determine the set of model parameters (i.e., initial abundance of each diagonal cohort and the annual rates of harvest mortality and natural survival) that best describes the observed data (Table 1). We compared the performance of four different numerical optimization methods available in Program R [30] to solve for the maximum likelihood of the objective function used in statistical population reconstruction. We implemented particle swarm optimization using the “pso” package [31] and a spectral projected gradient method [32] using the “BB” package [33]. We also used a bounded form of the Nelder–Mead algorithm [34] and a Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [35] using the base “stats” package in Program R. A conjugate gradient method [36] and a variant of simulated annealing [37], both using the base “stats” package, were also initially considered but were removed from the analysis because of excessive computational cost (mean execution times of several hours per simulation versus less than 40 s for the retained methods). Further, preliminary results consistently indicated that the precision and stability of these two methods was significantly lower than any of the four methods considered here.

This approach of numerical optimization of the parameter space allowed us to directly estimate natural survival (S), harvest vulnerability (C), initial age-class abundances in year 1 (N_{11}, N_{12}, N_{13}), and recruitment in subsequent years (N_{21}, \dots, N_{Y1}). All other year- and age-class specific abundances were estimated on the basis of the invariance property of the maximum likelihood estimation [1,23]:

$$N_{ij} = N_{i-1,j-1} e^{-C_{j-1} f_{i-1}} S_{j-1} \tag{6}$$

We calculated standard errors for the maximum likelihood estimates from a numerical estimate of the inverse Hessian [23,29,38] using the “numDeriv” package [39]. Because population reconstruction models consistently underestimate uncertainty [8], we inflated all standard errors by a goodness-of-fit scale parameter [1]:

$$\sqrt{\chi^2 / df} \tag{7}$$

where the χ^2 statistic is based on the observed age-at-harvest data (h_{ij}) and their expected values under the maximum likelihood solution to the objective function. The degrees of freedom (df) are equal to $A \times Y - K$, where A is the number of unique age-classes and K is the number of parameters estimated by the reconstruction model [1,3]. We then used these inflated standard errors to construct 95% confidence intervals for the model-derived estimates of yearly population abundance.

2.2. Simulation Study and Case Study

2.2.1. Simulation Study

We used a Monte Carlo simulation to determine the stability and precision of ML population reconstruction using each of the four alternative numerical optimization methods (Appendix A.1). This approach is frequently used to evaluate the performance of population reconstruction models [3,10,23,24,29,38] and provides the flexibility to simulate a wide range of demographic processes (e.g., harvest and recruitment) and parameters (e.g., low or high annual survival rates) that a population may experience. We developed a demographically stochastic version of a Leslie matrix model [40,41] in Program R to generate age-at-harvest data for a wildlife population with three distinct age classes (young-of-the-year, yearlings, and adults, as observed in the age-at-harvest matrix), exhibiting different levels of natural survival and harvest rates [3,23,24].

We generated recruitment for each simulated year using a Poisson process, with adjustments made to produce populations with stationary abundance in expectation that fluctuated as a result of stochasticity [3,10,23,24]. We modeled natural survival and harvest as binomial processes, determined stochastically using the probabilities of success (i.e., survival and harvest rates), each of which was drawn from the uniform distributions defined below. We generated catch effort by re-sampling with replacement the relative catch effort observed in Indiana over the last six years (Table 1).

For each simulation, we generated 30 years of data to establish demographic trends, with an additional six years used for the population reconstruction. To minimize the number of demographic scenarios investigated, we assumed that survival and harvest vulnerability did not vary by age-class [3,10,23,24]. Simulations were designed to represent a wide variety of harvested populations, with natural survival rates between 0.75 and 0.90, harvest rates between 0.10 and 0.25, and initial population sizes between 5000 and 15,000 individuals. These distributions and values were based on those used in previous simulation studies of statistical population reconstruction [3,10,23,24]. A total of 10,000 simulations were analyzed overall.

Using our simulated data, we compared the stability and precision of point estimates derived from population reconstruction using each of the four numerical optimization methods by calculating the mean and standard deviation of the relative absolute deviation (RAD) in annual abundance estimates for each scenario, defined as follows:

$$\text{RAD} = \frac{1}{Y} \sum_{i=1}^Y \frac{|\hat{N}_i - N_i|}{N_i}, \quad (8)$$

where N_i is the true simulated overall (i.e., not cohort-specific) abundance in year i and \hat{N}_i is the corresponding estimate derived by applying population reconstruction to the simulated age-at-harvest and catch-effort data [24]. We used paired t -tests to compare the performance of the four numerical optimization methods in terms of the relative absolute deviation of the best-fit model identified by each method, as well as in terms of the corresponding Akaike's Information Criterion (AIC) [42] for each model.

2.2.2. Case Study: Population Reconstruction of River Otter in Indiana

We examined four alternative models for the actual (i.e., not simulated) river otter population reconstruction, testing whether harvest vulnerability or natural survival were age-class-specific or age-class-constant (Appendix A.2). We used Akaike's Information Criterion (AIC) for model selection and compared the best-fit model under each of the four numerical optimization methods.

3. Results

3.1. Simulation Study

We simulated a variety of demographic scenarios a total of 10,000 times, reconstructing population abundance over a six-year period using four different numerical optimization

methods. Using particle swarm optimization to find the maximum of the objective function consistently yielded significantly better-fitting models (i.e., models with smaller AIC) when compared with the spectral projected gradient (mean $\Delta\text{AIC} = 0.814$; paired t -test, $t = 8.244$, $p < 0.001$), Nelder–Mead (mean $\Delta\text{AIC} = 2.518$, $t = 23.149$, $p < 0.001$), and Broyden–Fletcher–Goldfarb–Shanno (mean $\Delta\text{AIC} = 1.758$, $t = 17.550$, $p < 0.001$) methods.

The mean relative absolute deviation for these best-fit models using particle swarm optimization was 0.364 (SD = 0.169). The relative absolute deviation for the other three methods was significantly higher (indicating lower precision) in a manner largely consistent with the differences in AIC. Using the spectral projected gradient method resulted in the smallest loss of precision (mean $\Delta\text{RAD} = 0.070$, $t = 16.886$, $p < 0.001$), followed by larger losses when using the Nelder–Mead (mean $\Delta\text{RAD} = 0.402$, $t = 85.124$, $p < 0.001$) and Broyden–Fletcher–Goldfarb–Shanno (mean $\Delta\text{RAD} = 0.525$, $t = 149.470$, $p < 0.001$) methods.

3.2. Population Reconstruction of River Otter in Indiana

Using particle swarm optimization, the best available population reconstruction (i.e., the model with the smallest AIC) modeled both harvest vulnerability and natural survival as differing among young-of-the-year, yearlings, and adults (next best model $\Delta\text{AIC} = 6.476$). Using this model, we estimated a harvest vulnerability of 0.092 (SE = 0.011) for young-of-the-year, 0.140 (SE = 0.001) for yearlings, and 0.050 (SE = 0.001) for adults. Alongside annual changes in catch effort, these estimates resulted in average harvest rates decreasing from 10.3% in 2015 to 7.3% in 2018 before increasing back up to 9.1% by 2020. River otter abundance immediately before harvest showed a steady increase from 6506 (95% CI = 4957–8006) in 2015 to 8046 (6323–10,245) in 2020, with annual recruitment between 1530 (880–2192) and 2447 (1552–3199) animals (Figure 1).

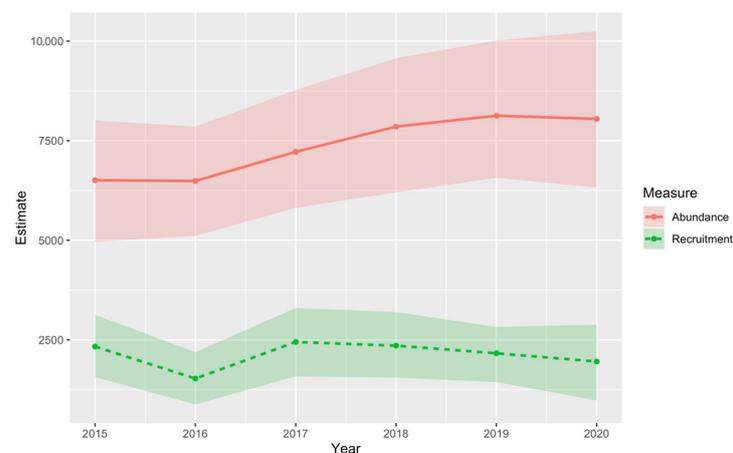


Figure 1. Estimated trends, along with associated 95% confidence intervals (shaded regions), in abundance (top line) and recruitment into the trappable population (bottom line) of North American river otter in Indiana between 2015 and 2020 based on the best available population reconstruction model using particle swarm optimization of an ML objective function.

Reconstruction using alternative numerical optimization methods also modeled both harvest vulnerability and natural survival as different for young-of-the-year, yearlings, and adults, but with substantially higher AIC (Table 2). This increase in AIC was consistent with our simulated results, with smaller increases when using the spectral projected gradient method ($\Delta\text{AIC} = 3.272$) and larger ones when using the Nelder–Mead ($\Delta\text{AIC} = 18.750$) and Broyden–Fletcher–Goldfarb–Shanno ($\Delta\text{AIC} = 42.973$) methods. Abundance estimates from these alternative models deviated from the estimates derived using the optimum approach (i.e., best-fit reconstruction using particle swarm optimization of the objective function) in a manner that was again consistent with our simulation results (i.e., larger deviations for reconstructions using the BFGS method, moderate ones using the Nelder–Mead method, and smaller ones using the spectral projected gradient method).

Table 2. Akaike Information Criterion (AIC) for alternative population reconstruction models of North American river otter in Indiana, USA, 2015–2020, using four different numerical optimization methods (particle swarm optimization (PSO), spectral projected gradient (SPG), Nelder–Mead, and Broyden–Fletcher–Goldfarb–Shanno (BFGS)). Models assumed harvest vulnerability (C) and natural survival (S) were either the same for all age-classes (subscript 1) or differed for young-of-the-year, yearlings, and adults (subscript 3).

Model	No. of Parameters	PSO	SPG	Nelder–Mead	BFGS
$M_{C_3S_3}$	14	540.493	543.765	559.243	583.466
$M_{C_3S_1}$	12	546.970	547.102	580.321	588.779
$M_{C_1S_3}$	12	701.840	754.976	729.224	744.487
$M_{C_1S_1}$	10	1216.019	1216.019	1216.041	1232.465

4. Discussion

Statistical population reconstruction using integrated population models provides a robust and flexible means of estimating population abundance and other demographic parameters across large spatial scales, which are necessary for effective management and conservation of harvested species of wildlife. Our simulation study, however, indicates that the precision and stability of reconstruction estimates can be strongly influenced by which numerical optimization method is used to estimate underlying model parameters. Methods such as the Nelder–Mead and Broyden–Fletcher–Goldfarb–Shanno algorithms, for example, are highly dependent on initial conditions and are more likely to converge to local rather than global optima. This can result in inaccurate estimates of natural survival and harvest vulnerability, which leads to substantial under- or over-estimation of overall population abundance. Interestingly, the spectral projected gradient method appears to be less dependent on initial conditions and consistently outperformed both the Nelder–Mead and Broyden–Fletcher–Goldfarb–Shanno methods in finding a more optimal solution in statistical population reconstruction. However, its dependence on initial conditions still results in it converging to a local, albeit more optimal, maximum, leading again to inaccurate estimates of underlying model parameters and corresponding population abundances.

Conversely, particle swarm optimization does not require initial guesses for model parameters, relying instead on a group of random particles that are populated and initialized throughout the multidimensional parameter space [14,16,18]. This independence from initial conditions not only decreases the odds of converging to a local rather than global optimum but also allows for optimization of objective functions that are discontinuous, as can occur in statistical population reconstruction. As expected, simulated reconstructions that used particle swarm optimization converged to significantly higher maximums than those that used the Nelder–Mead, Broyden–Fletcher–Goldfarb–Shanno, or spectral projected gradient methods. These results were consistent across a wide range of harvest intensities, natural survival rates, and population abundances. Given the heuristic nature of this method, it is not possible to guarantee that the precise global optimum had been found without an exhaustive and computationally infeasible search of the entire multidimensional parameter space. However, the consistency with which particle swarm optimization outperforms other numerical optimization methods strongly encourages its use for parameter estimation in statistical population reconstruction. These results further corroborate previous findings in other disciplines that particle swarm optimization provides a more effective means of finding optimum solutions in complex multidimensional space [18,19].

Our analysis also reaffirms previous findings that statistical population reconstructions based solely on age-at-harvest and catch-effort data may provide imprecise estimates of both cohort and overall abundance of wildlife populations [3,10,24], as indicated by a mean relative absolute deviation of 0.364 (SD = 0.169), even with the use of particle swarm optimization. Incorporating auxiliary data from radio-telemetry studies or independent estimates of annual abundance to help estimate one or more model parameters can sig-

nificantly improve the precision of these estimates [3,24], but these data are not always available (as was the situation with our case study). The amount of age-at-harvest data available (i.e., the number of animals harvested and aged each year) has also been shown to influence the precision of population reconstruction, particularly in the absence of auxiliary data [10]. As such, we continue to encourage managers to be cautious about statistically reconstructing populations where low abundance or low harvest rates do not generate a sufficient number of harvested animals each year (i.e., an average of at least 600 animals harvested per year [10]) unless other auxiliary data are available.

Use of statistical population reconstruction with the observed (i.e., not simulated) age-at-harvest and catch-effort data alongside particle swarm optimization suggests that the Indiana river otter population (measured pre-harvest) has been overall increasing over the past six years, albeit at a steadily slowing rate. Importantly, the number of animals harvested per year (mean = 573, range = 497 to 616) was slightly lower than that recommended by previous simulation studies [10], suggesting that these results should be treated with some caution. Use of alternative numerical optimization methods resulted in substantially lower estimates of harvest vulnerability and substantially higher estimates of overall population abundance, particularly when either the Nelder–Mead or Broyden–Fletcher–Goldfarb–Shanno algorithms were used. These results corroborate conclusions drawn from our simulation study and further demonstrate the need to avoid numerical optimization methods that rely heavily on initial conditions.

Statistical population reconstruction using particle swarm optimization estimated separate harvest vulnerability coefficients for each of the three age classes, with substantially higher coefficients for young-of-the-year and yearling than for adults (0.092 and 0.140, respectively, versus 0.050). This suggests that any changes or fluctuations in harvest pressure (e.g., through adjustments to bag limits or season lengths) are likely to have a larger impact on younger river otters than on older ones. These results are consistent with river otter reconstruction conducted in Kentucky and in Ohio, both of which also estimated higher harvest vulnerability for young-of-the-year and yearlings than for adults [10,43]. Interestingly, use of alternative numerical optimization methods did not change whether harvest vulnerability or natural survival should be modeled as age-class-specific or age-class-constant, suggesting that the structure of these estimates is somewhat robust to whether the model converges to a local or a global maximum.

Although the estimated overall abundance compares favorably with the results of independent population modeling conducted by the IN DNR (G. Albers, IN DNR, unpublished data), our reconstructed estimates indicate that annual recruitment of young-of-the-year into the trappable population actually has been declining over the past four years. This suggests that the observed increase in annual abundance is caused by decreasing mortality after recruitment rather than increasing reproductive success and pup survival prior to recruitment. This decrease in mortality after recruitment was likely caused by the sharp decrease in harvest pressure (i.e., probability of harvest) exerted on the river otter population at the start of this four-year period (from 9.9% in 2017 to 7.3% in 2018). As harvest pressure has steadily increased since 2018, however, the increase in annual abundance has slowed, finally reversing in 2020 when abundance decreased slightly from 8125 (95% CI = 6569–10,015) the previous year to 8046 (6323–10,245) animals. This decrease coincided with higher harvest pressure than either of the previous two years (9.1% in 2020 versus 7.3% and 8.6% in 2018 and 2019, respectively), suggesting that catch effort plays a large role in governing the growth of the river otter population in Indiana.

Whether the observed decline in annual recruitment is transient or indicative of a more persistent decrease in either reproductive success or pup survival is difficult to discern, particularly given the relatively small number of years with sufficient data for population reconstruction. Previous simulation studies have demonstrated that the stability and precision of reconstruction estimates increase dramatically as more years of age-at-harvest and catch-effort data are included, especially in the absence of auxiliary data [24]. Future reconstruction efforts should aim to include additional years of age-at-harvest and catch-

effort data as they become available, as well as explore the benefits of starting to collect other auxiliary data. Given the decreasing trend in annual recruitment, auxiliary data on annual pregnancy rates, litter size, or pup survival should be a priority. Incorporating such data, should they become available, may help to separate the effects of pup survival and reproductive success on annual recruitment, thereby allowing managers to better identify the driving forces behind observed population trends.

5. Conclusions

Although statistical population reconstruction can provide managers with an efficient means of monitoring the population trends of harvested species across large geographical areas, failure to ensure model convergence to a global rather than a local maximum can result in imprecise results. Unfortunately, many of the numerical optimization methods that may be used in population reconstruction models are unlikely to converge to a global maximum because of their dependence on initial conditions and may, therefore, lead to significant overestimation of population abundance. Given that these estimates are often used to guide management and conservation of harvested wildlife populations, we encourage management agencies to treat any results derived from reconstruction based on such methods with caution. Our results do suggest that using particle swarm optimization in place of other optimization methods consistently results in convergence to a higher maximum and, as such, can significantly improve the stability and precision of reconstruction estimates. This is particularly important if other auxiliary information from index surveys, mark–recapture analyses, or radio-telemetry studies is not available, which may already result in imprecise estimates of population abundance. As such, we encourage management agencies to incorporate particle swarm optimization alongside other numerical optimization methods already used in reconstruction models to determine optimal bag limits and season lengths for wildlife species that are in high demand by the public.

Funding: This material is based upon work supported by a Wildlife Restoration Grant in cooperation with the Indiana Department of Natural Resources, Division of Fish & Wildlife, and the U.S. Fish & Wildlife Service Wildlife Restoration Grant Program under Grant Number F21AF02467, W-51-R-02.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to G. Albers and E. McCallen for contributions and suggestions at every stage. River otter data were provided by the Indiana Department of Natural Resources. We especially want to thank the wildlife technicians who collected and processed teeth from harvested animals to estimate the age at harvest.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Excerpts of Code Written in Program R to Perform Both the Simulation Study and the Case Study of Population Reconstruction

Appendix A.1. Code Used to Perform Simulation Study

```
#####
###                                                                                                     ###
###                                                                                                     ###
###                                                                                                     ###
###                                                                                                     ###
### Excerpt of code written in Program R to evaluate effects of  ###
### alternative numerical optimization methods on the precision  ###
### of maximum likelihood statistical population reconstruction  ###
###                                                                                                     ###
#####
#####
##### Set Up Work Space #####
#####
```

```

# Clear global environment, import required packages, and set working directory
{
  rm(list=ls())

  require(doSNOW) # For multi-core processing
  require(truncnorm); require(numDeriv); require(pso); require(BB)
}

#####
##### Define Multi-Core Simulation #####
#####

simulateReconstruction <- function() {

  # Define Monte Carlo simulation
  simulatePopulation <- function(par) {

    ## Initialize matrix to record simulated harvest counts
    {
      h <- matrix(0, nrow = 30 + Y, ncol = A)
    }

    ## Expand effort estimates for establishment and reconstruction years
    {
      f <- c(f0[sample(1:Y, size = 30, replace = TRUE)],
            f0[sample(1:Y, size = Y, replace = FALSE)])
      f <- matrix(rep(f, A), ncol = A)
    }

    ## Define initial (diagonal) cohort values
    {
      N <- matrix(NA, nrow = 30 + Y, ncol = A)
      N[1, 1:A] = round(n / 3)
    }

    ## Define vulnerability and survival values
    {
      c <- matrix(c, nrow = 30 + Y, ncol = 3)
      s <- matrix(s, nrow = 30 + Y, ncol = 3)
    }

    ## Define probability of harvest
    {
      P <- (1 - exp(-c * f))
    }

    ## Define population sizes and harvest counts for simulated years
    {
      for (i in 1:(30 + Y - 1)) {

        ### Simulate harvest
        {
          h[i,] <- rbinom(rep(1, A), N[i,] , P[i,])
        }

        ### Simulate survival
        {
          for (j in 1:(A - 2)) {
            N[i+1, j+1] <- rbinom(1, N[i, j] - h[i, j], s[i, j])
          }
        }
      }
    }
  }
}

```

```

        for (j in (A - 1)) {
            N[i+1, j+1] <- rbinom(1, N[i, j] - h[i, j], s[i, j]) +
                rbinom(1, N[i, j + 1] - h[i, j + 1], s[i, j + 1])
        }

        N[i+1, 1] <- round((sum(N[i, 1:A]) - sum(N[i + 1, 2 : A])) *
            runif(1, min = 0.80, max = 1.25))
    }
}

## Simulate harvest and index survey during the final year
{
    h[(30 + Y), ] <- rbinom(rep(1, A), N[(30 + Y), ], P[(30 + Y), ])
}

# Trim establishment years from the simulated data
{
    N <- N[-c(1:30), ]
    h <- h[-c(1:30), ]
    f = f[-c(1:30), ]
}

# Return population simulated values
{
    return(list(N = N, h = h, f = f))
}

# Define likelihood variant of the objective function
objectiveFunction <- function(par) {

    ## Import initial (diagonal) cohort values
    {
        N <- matrix(NA, nrow = Y, ncol = A)
        N[1, 1:A] <- par[1:A]
        N[2:Y, 1] <- par[(A + 1):(A + Y - 1)]
    }

    ## Import vulnerability and survival estimates
    {
        C <- matrix(par[Y + (A - 1) + c(1, 1, 1)] / scaleFactor,
            nrow = Y, ncol = A, byrow = T)
        S <- matrix(par[Y + (A - 1) + c(2, 2, 2)] / scaleFactor,
            nrow = Y, ncol = A, byrow = T)
    }

    ## Define probability of harvest
    {
        P <- (1 - exp(-C * f))
    }

    ## Define expected population sizes
    {
        for (i in 1:(Y - 1)) {
            for (j in 1:(A - 2)) {
                N[i + 1, j + 1] <- N[i, j] * (1 - P[i, j]) * S[i, j]
            }

            for (j in (A - 1)) {

```

```

        N[i + 1, j + 1] <- N[i, j] * (1 - P[i, j]) * S[i, j] +
                               N[i, j + 1] * (1 - P[i, j + 1]) * S[i, j + 1]
    }
}
}
# Return population size estimate (if requested)
{
  if (returnPopulationAbundance) return (N)
}
# Perform initial test of model boundaries
{
  if (any(N < h)) return (9000009)
  if (any(C < limitForC[1],
          C > limitForC[2])) return (9000008)
  if (any(S < limitForS[1],
          S > limitForS[2])) return (9000007)
  if (any(P + (1 - S) > 0.500)) return (9000006)
}
# Define expected harvest values
{
  E <- N * P
}
## Return chi-square estimate for inflation factor if requested
{
  if (returnChiSquareCorrection) {
    return (sum((h - E) ^ 2 / E))
  }
}
## Calculate contributions of each likelihood component
{
  ### Initialize the age-at-harvest and catch-effort components
  {
    logL_AAH <- matrix(0, nrow = Y, ncol = A)
    logL_CAE <- matrix(0, nrow = Y, ncol = A)
  }
  ### Contribution of the age-at-harvest component
  {
    logL_AAH[1, A] <- sum(log((N[1,A] + 1 - 1:h[1,A]) / 1:h[1,A])) +
                    (h[1,A] * (log(P[1,A])) +
                     (N[1,A] - h[1,A]) * (log(1 - P[1,A])))
    logL_AAH[1, (A - 1)] <- sum(log((h[1,2] + h[2,3] + 1 - 1:h[1,2]) /
                                   1:h[1,2])) +
                          h[1,2] * log(E[1,2] / (E[1,2] + E[2,3])) +
                          h[2,3] * log(E[2,3] / (E[1,2] + E[2,3]))
    for (i in 1 : (Y - 2)) {
      logL_AAH[i, 1] <- sum(log((h[i,1] + h[i+1,2] + h[i+2,3] +
                                1 - 1:h[i,1]) / 1:h[i,1])) +
                        sum(log((h[i+1,2] + h[i+2,3] +
                                1 - 1:h[i+1,2]) / 1:h[i+1,2])) +
                        h[i+0,1] * log(E[i+0,1] / (E[i,1] + E[i+1,2] + E[i+2,3])) +
                        h[i+1,2] * log(E[i+1,2] / (E[i,1] + E[i+1,2] + E[i+2,3])) +

```

```

      h[i+2,3] * log(E[i+2,3] / (E[i,1] + E[i+1,2] + E[i+2,3]))
    }
  logL_AAHA[(Y - 1), 1] <- sum(log((N[Y-1,1]
    +
    1 - 1:h[Y-1,1]) / 1:h[Y-1,1])) +
    sum(log((N[Y-1,1] - h[Y-1,1] +
    1 - 1:h[Y,2]) / 1:h[Y,2])) +
    (h[Y-1,1]) * log((P[Y-1,1])) +
    (h[Y,2]) * log((1 - P[Y-1,1]) *
    S[1] * P[Y,2]) +
    (N[Y-1,1] - h[Y-1,1] - h[Y,2]) * log((1 - P[Y-1,1]) *
    (1 - S[1] * P[Y,2]))
  logL_AAHA[Y, 1] <- sum(log((N[Y,1] + 1 - 1:h[Y,1]) / 1:h[Y,1])) +
    (h[Y,1]) * (log(P[Y,1])) +
    (N[Y,1] - h[Y,1]) * (log(1 - P[Y,1]))
}
### Contribution of the catch-effort component
{
  for (i in 1:Y) {
    for (j in 1:A) {
      logL_CAE[i,j] <- sum(log((sum(N[i,j]) + 1 - 1:sum(h[i,j])) /
        1:sum(h[i,j]))) +
        sum(h[i,j]) * (log(P[i,j])) +
        sum(N[i,j] - h[i,j]) * (log(1 - P[i,j]))
    }
  }
}
## Return value of objective function if valid
{
  logLikelihood <- -c(logL_AAHA, logL_CAE)
  if (any(is.na(logLikelihood))) return (9000003) else
  if (any(logLikelihood == -Inf)) return (9000002) else
  if (any(logLikelihood == Inf)) return (9000001) else
  return(sum(logLikelihood))
}
}
# Define model parameterizations
{
  ## Import estimates of yearly catch-effort
  {
    f0 <- c(1.1552360, 0.9585951, 1.1134204, 0.8088820, 0.9529701, 1.0108965)
  }
  ## Define number of years and age classes
  {
    Y <- length(f0)
    A <- 3
  }
  ## Set reasonable limits for survival and harvest vulnerability
  {
    limitForC <- c(0.05, 0.50)
    limitForS <- c(0.50, 0.95)
  }
}

```



```

    }
  }

  ## Check for validity of results
  if (!(any(is.na(result$RAD)))) {
    break()
  }
}

# Return validated results
return(result)
}

#####
##### Perform Numerical Optimization of Simulated Results #####
#####

# Initialize multi-core processing
{
  maxIterations <- 1000
  core <- 10
  cl <- makeCluster(core, type = "SOCK")
  registerDoSNOW(cl)
}

# Initialize data frame to store final results
{
  finalResults <- data.frame(VLN = NA, SRV = NA, ABN = NA, HRV = NA,
                             ALG = NA, RAD = NA, AIC = NA)
}

# Cycle through sufficient number of simulations
{
  for (counter in 1:ceiling(maxIterations / core)) {

    ## Simulate reconstruction
    {
      results <- foreach(i = 1:core, .combine = rbind,
                        .packages = c("readxl","pso","BB",
                                      "numDeriv","truncnorm"),
                        .errorhandling = "remove") %dopar% (
        simulateReconstruction()
      )
    }

    ## Append simulated results to data frame
    {
      finalResults <- rbind(finalResults, as.data.frame(results))
      finalResults <- finalResults[complete.cases(finalResults), ]
    }

    ## Update and export partial results
    {
      print(paste(round(100 * nrow(finalResults) / maxIterations /
                    (nrow(results) / core), 2),
                  "% at ", Sys.time(), sep = ""))
      write.csv(finalResults, "Simulated Results for IN Otters MK1v1.csv",
                row.names = FALSE)
    }

    ## Visualize partial results

```



```

        174, 260, 182), byrow = T, nrow = 6, ncol = 3)
    }

# Import estimates of yearly catch-effort
{
  f <- c(1.1552360, 0.9585951, 1.1134204, 0.8088820, 0.9529701, 1.0108965)
}

## Define number of years and age classes
{
  Y <- nrow(h)
  A <- ncol(h)
  yearRange <- c(2015, 2016, 2017, 2018, 2019, 2020)
}

# Set reasonable limits for survival and harvest vulnerability
{
  limitForC <- c(0.05, 0.50)
  limitForS <- c(0.50, 0.95)
}

#####
##### Define Objective Function #####
#####

# Define objective function using a multinomial likelihood formulation
objectiveFunction <- function(par) {

  ## Import initial (diagonal) cohort values
  {
    N <- matrix(NA, nrow = Y, ncol = A)
    N[1, 1:A] <- par[1:A]
    N[2:Y, 1] <- par[(A + 1):(A + Y - 1)]
  }

  ## Import vulnerability and survival estimates
  {
    if (vulnerability == 1) {
      C <- matrix(par[Y + (A - 1) + c(1, 1, 1)] / scaleFactor,
                 nrow = Y, ncol = A, byrow = T)
    }
    if (vulnerability == 2) {
      C <- matrix(par[Y + (A - 1) + c(1, 2, 2)] / scaleFactor,
                 nrow = Y, ncol = A, byrow = T)
    }
    if (vulnerability == 3) {
      C <- matrix(par[Y + (A - 1) + c(1, 2, 3)] / scaleFactor,
                 nrow = Y, ncol = A, byrow = T)
    }
    if (survivability == 1) {
      S <- matrix(par[Y + (A - 1) + c(4, 4, 4)] / scaleFactor,
                 nrow = Y, ncol = A, byrow = T)
    }
    if (survivability == 2) {
      S <- matrix(par[Y + (A - 1) + c(4, 5, 5)] / scaleFactor,
                 nrow = Y, ncol = A, byrow = T)
    }
    if (survivability == 3) {

```

```

        S <- matrix(par[Y + (A - 1) + c(4, 5, 6)] / scaleFactor,
                  nrow = Y, ncol = A, byrow = T)
    }
}

## Define probability of harvest
{
  P <- (1 - exp(-C * f))
}

## Define expected population sizes
{
  for (i in 1:(Y - 1)) {
    for (j in 1:(A - 2)) {
      N[i + 1, j + 1] <- N[i, j] * (1 - P[i, j]) * S[i, j]
    }
    for (j in (A - 1)) {
      N[i + 1, j + 1] <- N[i, j] * (1 - P[i, j]) * S[i, j] +
        N[i, j + 1] * (1 - P[i, j + 1]) * S[i, j + 1]
    }
  }
}

# Return population size estimate (if requested)
{
  if (returnPopulationAbundance) return (N)
}

# Perform initial test of model boundaries
{
  if (any(N < h)) return (9000009)
  if (any(C < limitForC[1],
          C > limitForC[2])) return (9000008)
  if (any(S < limitForS[1],
          S > limitForS[2])) return (9000007)
  if (any(P + (1 - S) > 0.500)) return (9000006)
}

# Define expected harvest values
{
  E <- N * P
}

## Return chi-square estimate for inflation factor if requested
{
  if (returnChiSquareCorrection) {
    return (sum((h - E) ^ 2 / E))
  }
}

## Calculate contributions of each likelihood component
{
  ### Initialize the age-at-harvest and catch-effort components
  {
    logL_AAH <- matrix(0, nrow = Y, ncol = A)
    logL_CAE <- matrix(0, nrow = Y, ncol = A)
  }
}

```

```

### Contribution of the age-at-harvest component
{
  logL_AAHA[1, A] <- sum(log((N[1,3] + 1 - 1:h[1,3]) / 1:h[1,3])) +
    h[1,3] * (log(P[1,A])) +
    (N[1,3] - h[1,3]) * (log(1 - P[1,A]))

  logL_AAHA[1, (A - 1)] <- sum(log((h[1,2] + h[2,3] + 1 - 1:h[1,2]) /
    1:h[1,2])) +
    h[1,2] * log(E[1,2] / (E[1,2] + E[2,3])) +
    h[2,3] * log(E[2,3] / (E[1,2] + E[2,3]))

  for (i in 1 : (Y - 2)) {
    logL_AAHA[i, 1] <- sum(log((h[i,1] + h[i+1,2] + h[i+2,3] +
      1 - 1:h[i,1]) / 1:h[i,1])) +
      sum(log((h[i+1,2] + h[i+2,3] +
        1 - 1:h[i+1,2]) / 1:h[i+1,2])) +
      h[i+0,1] * log(E[i+0,1] / (E[i,1] + E[i+1,2] + E[i+2,3])) +
      h[i+1,2] * log(E[i+1,2] / (E[i,1] + E[i+1,2] + E[i+2,3])) +
      h[i+2,3] * log(E[i+2,3] / (E[i,1] + E[i+1,2] + E[i+2,3]))
  }

  logL_AAHA[(Y - 1), 1] <- sum(log((N[Y-1,1] +
    1 - 1:h[Y-1,1]) / 1:h[Y-1,1])) +
    sum(log((N[Y-1,1] - h[Y-1,1] +
      1 - 1:h[Y,2]) / 1:h[Y,2])) +
    (h[Y-1,1]) * log(P[Y-1,1])) +
    (h[Y,2]) * log((1 - P[Y-1,1]) *
      S[1] * P[Y,2]) +
    (N[Y-1,1] - h[Y-1,1] - h[Y,2]) * log((1 - P[Y-1,1]) *
      (1 - S[1] * P[Y,2]))

  logL_AAHA[Y, 1] <- sum(log((N[Y,1] + 1 - 1:h[Y,1]) / 1:h[Y,1])) +
    (h[Y,1]) * (log(P[Y,1])) +
    (N[Y,1] - h[Y,1]) * (log(1 - P[Y,1]))
}

### Contribution of the catch-effort component
{
  for (i in 1:Y) {
    for (j in 1:A) {
      logL_CAE[i,j] <- sum(log((sum(N[i,j]) + 1 - 1:sum(h[i,j])) /
        1:sum(h[i,j]))) +
        sum(h[i,j]) * (log(P[i,j])) +
        sum(N[i,j] - h[i,j]) * (log(1 - P[i,j]))
    }
  }
}

## Return value of objective function if valid
{
  logLikelihood <- -c(logL_AAHA, logL_CAE)

  if (any(is.na(logLikelihood))) return (9000003) else
  if (any(logLikelihood == -Inf)) return (9000002) else
  if (any(logLikelihood == Inf)) return (9000001) else
  return(sum(logLikelihood))
}

```

```

}

#####
##### Perform NUMERICAL OPTIMIZATION of Observed Data #####
#####

# Initialize data frames to store results
{
  pointEstimatesPSO <- cbind(data.frame(N11 = rep(NA, 4), N12 = NA, N13 = NA),
                             data.frame(matrix(NA, nrow = 4, ncol = (Y - 1))),
                             data.frame(C1 = rep(NA, 4), C2 = NA, C3 = NA,
                                         S1 = NA, S2 = NA, S3 = NA, VAL = NA))
  populationSizePSO <- data.frame(matrix(NA, nrow = 4, ncol = Y))

  pointEstimatesSPG <- pointEstimatesPSO
  pointEstimatesNLM <- pointEstimatesPSO
  pointEstimatesBFG <- pointEstimatesPSO

  populationSizeSPG <- populationSizePSO
  populationSizeNLM <- populationSizePSO
  populationSizeBFG <- populationSizePSO
}

# Initialize starting values for parameterization
{
  scaleFactor <- 1e5
  initialValues <- c(rep(c(mean(h) * 10), Y + (A - 1)),
                    rep(c(0.10, 0.70), each = A) * scaleFactor)
}

# Optimize parameter space using four different numerical optimization methods
{
  ## Initialize model counter
  {
    modelCount <- 1
  }

  ## Cycle through possible vulnerability and survival delineations
  for (vulnerability in c(1,3)) {
    for (survivability in c(1,3)) {

      ### Optimize objective function using particle swarm optimization
      {
        returnPopulationAbundance <- FALSE
        returnChiSquareCorrection <- FALSE
        optimized <- psoptim(par = initialValues,
                             fn = objectiveFunction,
                             lower = c(rep(c(00000), A + Y - 1),
                                       rep(limitForC[1] * scaleFactor, A),
                                       rep(limitForS[1] * scaleFactor, A)),
                             upper = c(rep(c(20000), A + Y - 1),
                                       rep(limitForC[2] * scaleFactor, A),
                                       rep(limitForS[2] * scaleFactor, A)),
                             control = list(maxit = 10000, trace = F,
                                           hybrid = "improved",
                                           vectorize = TRUE,
                                           type = "SPSO2011"))

        returnPopulationAbundance <- TRUE
      }
    }
  }
}

```



```

        populationSizeBFG[modelCount, ] <- rowSums(estimatedN)
        returnPopulationAbundance <- FALSE
    }

    ### Increment model counter
    {
        modelCount <- modelCount + 1
    }
}

## Compute AIC for each reconstruction estimate
{
    pointEstimatesPSO$AIC <- 2 * pointEstimatesPSO$VAL +
        2 * ((A + Y - 1) + c(2, 4, 4, 6))
    pointEstimatesSPG$AIC <- 2 * pointEstimatesSPG$VAL +
        2 * ((A + Y - 1) + c(2, 4, 4, 6))
    pointEstimatesNLM$AIC <- 2 * pointEstimatesNLM$VAL +
        2 * ((A + Y - 1) + c(2, 4, 4, 6))
    pointEstimatesBFG$AIC <- 2 * pointEstimatesBFG$VAL +
        2 * ((A + Y - 1) + c(2, 4, 4, 6))
}

# Extract uncertainty estimates for best-fit model
{
    ## Construct data frame to compile subsequent results
    {
        bestFitModel <- data.frame(Year = yearRange,
            Estimate = NA,
            Lo95CI = NA, Hi95CI = NA,
            Measure = rep(c("Abundance",
                "Recruitment"),
                each = length(yearRange)))
    }

    ## Calculate standard errors for best-fit model parameters
    {
        bestParameters <- as.numeric(pointEstimatesPSO[4, 1:(A + Y - 1 + A + A)])
        vulnerability <- 3
        survivability <- 3

        returnPopulationAbundance <- FALSE
        hessianMatrix <- abs(hessian(x = bestParameters,
            func = objectiveFunction))

        standardErrors <- sqrt(abs(diag(solve(hessianMatrix))))

        returnChiSquareCorrection <- TRUE
        chiSquare <- objectiveFunction(bestParameters)
        returnChiSquareCorrection <- FALSE

        standardErrors <- standardErrors * sqrt(chiSquare / length(bestParameters))
    }

    ## Calculate stochastic abundance estimates using standard errors
    {
        abundanceProjection <- matrix(NA, nrow = 1000, ncol = Y)
        recruiterProjection <- matrix(NA, nrow = 1000, ncol = Y)
        tempParameters <- bestParameters
    }
}

```

```

returnPopulationAbundance <- TRUE
for (i in 1:nrow(abundanceProjection)) {
  for (j in 1:(A + Y - 1)) {
    tempParameters[j] <- rtruncnorm(1, a = 0,
                                   mean = bestParameters[j],
                                   sd = standardErrors[j])
  }
  for (j in (A + Y - 1 + 1:(2 * A))) {
    tempParameters[j] <- rtruncnorm(1, a = 0, b = scaleFactor,
                                   mean = bestParameters[j],
                                   sd = standardErrors[j])
  }

  returnPopulationAbundance <- TRUE
  abundanceProjection[i,] <- rowSums(objectiveFunction(tempParameters))
  recruiterProjection[i,] <- (objectiveFunction(tempParameters))[,1]
}
}

## Compute confidence intervals
{
  bestFitModel$Estimate[bestFitModel$Measure == "Abundance"] <-
    as.numeric(populationSizePSO[4,])

  bestFitModel$Lo95CI[bestFitModel$Measure == "Abundance"] <-
    apply(abundanceProjection, 2, quantile, prob = 0.025)

  bestFitModel$Hi95CI[bestFitModel$Measure == "Abundance"] <-
    apply(abundanceProjection, 2, quantile, prob = 0.975)

  bestFitModel$Estimate[bestFitModel$Measure == "Recruitment"] <-
    as.numeric(pointEstimatesPSO[4,c(1, (A - 1) + c(2:Y))])

  bestFitModel$Lo95CI[bestFitModel$Measure == "Recruitment"] <-
    apply(recruiterProjection, 2, quantile, prob = 0.025)

  bestFitModel$Hi95CI[bestFitModel$Measure == "Recruitment"] <-
    apply(recruiterProjection, 2, quantile, prob = 0.975)
}
}

#####
##### Construct VISUALIZATION Best-Fit Reconstruction #####
#####

# Generate figure of abundance and recruitment estimates
{
  ggplot(aes(y = Estimate, x = Year, fill = Measure),
         data = bestFitModel) +
    geom_ribbon(aes(ymin = Lo95CI, ymax = Hi95CI,
                  fill = Measure), alpha = 0.25) +
    geom_point(aes(colour = Measure)) +
    geom_path(aes(colour = Measure, linetype = Measure), size = 1) +
    scale_x_continuous(breaks = seq(min(yearRange), max(yearRange), 1)) +
    scale_fill_manual(values = c("#F8766D", "#00BA38")) +
    scale_color_manual(values = c("#F8766D", "#00BA38"))
}

#####

```


3. Berg, S.S.; Erb, J.D.; Fieberg, J.R.; Forester, J.D. Utility of radio-telemetry data for improving statistical population reconstruction. *J. Wildl. Manag.* **2017**, *81*, 535–544. [[CrossRef](#)]
4. Severud, W.J.; Berg, S.S.; Ernst, C.A.; DelGiudice, G.D.; Moore, S.A.; Windels, S.K.; Moen, R.A.; Isaac, E.J.; Wolf, T.M. Statistical population reconstruction of moose (*Alces alces*) in northeastern Minnesota using integrated population models. *PLoS ONE* **2022**, *17*, e0270615. [[CrossRef](#)]
5. Hatter, I.W.; Mowat, G.; McLellan, B.N. Statistical population reconstruction to evaluate grizzly bear trends in British Columbia, Canada. *Ursus* **2018**, *29*, 1. [[CrossRef](#)]
6. Sturludottir, E.; Nielsen, O.K.; Stefansson, G. Evaluation of ptarmigan management with a population reconstruction model. *J. Wildl. Manag.* **2018**, *82*, 958–965. [[CrossRef](#)]
7. Schaub, M.; Abadi, F. Integrated population models: A novel analysis framework for deeper insights into population dynamics. *J. Ornithol.* **2011**, *152*, 227–237. [[CrossRef](#)]
8. Gast, C.M. Fixed and Random Effects Models and Multistage Estimation Procedures for Statistical Population Reconstructions. Ph.D. Thesis, University of Washington, Seattle, WA, USA, 2012.
9. Berg, S.S. Modeling and Conservation of Wildlife Populations in Managed Landscapes: A Trade-Off Between Effort and Results. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, USA, 2016.
10. Berg, S.S.; Palmer, L.L. A comparison of multinomial likelihood and chi-square approaches to statistical population reconstruction. *J. Biol. Syst.* **2021**, *29*, 543–559. [[CrossRef](#)]
11. Gove, N.E.; Skalski, J.R.; Zager, P.; Townsend, R.L. Statistical models for population reconstruction using age-at-harvest data. *J. Wildl. Manag.* **2002**, *66*, 310–320. [[CrossRef](#)]
12. Broms, K.; Skalski, J.R.; Millspaugh, J.J.; Hagen, C.A.; Schulz, J.H. Using statistical population reconstruction to estimate demographic trends in small game populations. *J. Wildl. Manag.* **2010**, *74*, 310–317. [[CrossRef](#)]
13. Clawson, M.V.; Skalski, J.R.; Lady, J.M.; Hagen, C.A.; Millspaugh, J.J.; Budeau, D.; Severson, J.P. Performing statistical population reconstruction using Program PopRecon 2.0: Program PopRecon. *Wildl. Soc. Bull.* **2017**, *41*, 581–589. [[CrossRef](#)]
14. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
15. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
16. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]
17. Zhang, Y.; Wang, S.; Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Math. Probl. Eng.* **2015**, *2015*, 931256. [[CrossRef](#)]
18. Schwaab, M.; Biscaia, E.C., Jr.; Monteiro, J.L.; Pinto, J.C. Nonlinear parameter estimation through particle swarm optimization. *Chem. Eng. Sci.* **2008**, *63*, 1542–1552. [[CrossRef](#)]
19. Zhu, H.; Wang, Y.; Wang, K.; Chen, Y. Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. *Expert Syst. Appl.* **2011**, *38*, 10161–10169. [[CrossRef](#)]
20. Berg, S. Using demographically stochastic modeling to study the effects of cub survival on Amur leopard population trends. In Proceedings of the 2nd Stochastic Modeling Techniques and Data Analysis International Conference, Chania, Greece, 5–8 June 2012; Volume 5, pp. 37–42.
21. Caswell, H. *Matrix Population Models: Construction, Analysis, and Interpretation*, 2nd ed.; Sinauer Associates, Inc.: Sunderland, MA, USA, 2001.
22. Abadi, F.; Gimenez, O.; Ullrich, B.; Arlettaz, R.; Schaub, M. Estimation of immigration rate using integrated population models. *J. Appl. Ecol.* **2010**, *47*, 393–400. [[CrossRef](#)]
23. Skalski, J.R.; Millspaugh, J.J.; Clawson, M.V. Comparison of statistical population reconstruction using full and pooled adult age-class data. *PLoS ONE* **2012**, *7*, e33910. [[CrossRef](#)]
24. Clawson, M.V.; Skalski, J.R.; Millspaugh, J.J. The utility of auxiliary data in statistical population reconstruction. *Wildl. Biol.* **2013**, *19*, 147–155. [[CrossRef](#)]
25. Quinn, T.J.; Deriso, R.B. *Quantitative Fish Dynamics*; Oxford University Press: New York, NY, USA, 1999; ISBN 978-0-19-507631-8.
26. Seber, G.A.F. *Estimation of Animal Abundance and Related Parameters*, 2nd ed.; Blackburn Press: Caldwell, NJ, USA, 2002.
27. Skalski, J.R.; Millspaugh, J.J.; Clawson, M.V.; Belant, J.L.; Etter, D.R.; Frawley, B.J.; Friedrich, P.D. Abundance trends of American martens in Michigan based on statistical population reconstruction. *J. Wildl. Manag.* **2011**, *75*, 1767–1773. [[CrossRef](#)]
28. Skalski, J.R.; Clawson, M.V.; Millspaugh, J.J. Model evaluation in statistical population reconstruction. *Wildl. Biol.* **2012**, *18*, 225–234. [[CrossRef](#)]
29. Gast, C.M.; Skalski, J.R.; Isabelle, J.L.; Clawson, M.V. Random effects models and multistage estimation procedures for statistical population reconstruction of small game populations. *PLoS ONE* **2013**, *8*, e65244. [[CrossRef](#)] [[PubMed](#)]
30. R Core Team. *R: A Language and Environment for Statistical Computing*; R Core Team: Vienna, Austria, 2022.
31. Bendtsen, C. *ps: Particle Swarm Optimization. R Package*; Version 1.0.4; R Core Team: Vienna, Austria, 2022.
32. Birgin, E.G.; Martínez, J.M.; Raydan, M. Nonmonotone Spectral Projected Gradient Methods on Convex Sets. *SIAM J. Optim.* **2000**, *10*, 1196–1211. [[CrossRef](#)]

33. Varadhan, R.; Gilbert, P. BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. *J. Stat. Softw.* **2009**, *32*, 1–26. [[CrossRef](#)]
34. Nelder, J.A.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308–313. [[CrossRef](#)]
35. Shanno, D.F. Conditioning of Quasi-Newton Methods for Function Minimization. *Math. Comp.* **1970**, *24*, 647–656. [[CrossRef](#)]
36. Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149–154. [[CrossRef](#)]
37. Bélisle, C. Convergence Theorems for a class of simulated annealing algorithms on Rd. *J. Appl. Probl.* **1992**, *29*, 885–895. [[CrossRef](#)]
38. Gast, C.; Skalski, J.R.; Beyer, D.E. Evaluation of fixed- and random-effects models and multistage estimation procedures in statistical population reconstruction. *J. Wildl. Manag.* **2013**, *77*, 1258–1270. [[CrossRef](#)]
39. Gilbert, P.; Varadhan, R. *numDeriv: Accurate Numerical Derivatives*. R package; Version 2016.8-1.1; R Core Team: Vienna, Austria, 2019.
40. Leslie, P. On the use of matrices in certain population mathematics. *Biometrika* **1945**, *33*, 183–212. [[CrossRef](#)]
41. Leslie, P. Some further notes on the use of matrices in populations mathematics. *Biometrika* **1948**, *35*, 213–245. [[CrossRef](#)]
42. Burnham, K.P.; Anderson, D.R. *Model Selection and Multimodel Inference*; Springer: New York, NY, USA, 2002.
43. Ellington, E.H.; Flournoy, P.D.; Dwyer, C.P.; Witt, M.D.; Gehrt, S.D. Assessment of river otter abundance following reintroduction. *Wildl. Res.* **2018**, *45*, 490–499. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.