

Constraint Optimisation Approaches for Designing Group-Living Captive Breeding Programmes

Matthew Forshaw^{1,2}, Rachel Gray¹, Bridgett vonHoldt³, Alexander Ochoa⁴, Joshua M. Miller⁵, Kristin E. Brzeski⁶, Adalgisa Caccone⁴, Evelyn L. Jensen¹

¹Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

²The Alan Turing Institute, London, NW1 2DB, United Kingdom

³Princeton University, Princeton, NJ 08544, USA

⁴Yale University, New Haven, CT 06520, USA

⁵MacEwan University, Edmonton, Canada

⁶Michigan Technological University, Michigan, USA

{matthew.forshaw, R.Gray2, Evelyn.Jensen}@ncl.ac.uk, mforshaw@turing.ac.uk, vonholdt@princeton.edu, alexander.ochoa@yale.edu, millerj253@macewan.ca, kbrzeski@mtu.edu, adalgisa.caccone@yale.edu

Abstract

Captive breeding programs play a critical role in combating the ongoing biodiversity crisis by preserving the most endangered species and supporting reintroduction efforts. Maintaining the genetic health of captive populations requires careful management to prevent inbreeding and maximize the effective population size. Decisions about which males and females should be bred together are guided by the principle of minimizing relatedness between pairs. Methods to select breeding pairs are well developed, however, some species' ecology requires them to live in groups, and evaluating optimal groupings of multiple males and females that would be suitable to breed together is a more complex problem. Current computational tools to support the design of group-living captive breeding programs suffer from challenges of scalability and flexibility. In this paper we demonstrate the applicability of constraint programming (CP) approaches to optimize breeding groups to minimize relatedness. We present the example of the Galápagos giant tortoises as the test case used to develop our approach. Exploration of the needs of this captive breeding program has informed the development of our flexible approach to capture the constraints on viable captive breeding program design. Our findings have directly informed the implementation of new group configurations at the captive breeding centre. We further demonstrate that our approach is broadly applicable in other contexts through a second case study, providing multi-objective optimisation of a breeding program of canids. Through these case studies and an ablation study using synthetic datasets, we show that our constraint optimisation approach provides an expressive and generalizable means to support captive breeding program design, including scaling to large captive populations, which are currently intractable using current computational methods.

Code — <https://github.com/MattForshaw/AAAI25-CaptiveBreeding>

1 Introduction

Globally there have been precipitous declines in wildlife populations and heightened rates of extinction, termed the

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Figure 1: Galápagos giant tortoises at the captive breeding center in Puerto Ayora, Galápagos.

Biodiversity Crisis (Western 1992). Captive breeding is playing an increasingly important role in preserving species at immediate risk of extinction, often with the goal of captive bred individuals being released to supplement depleted wild populations or reintroduce species where they have been extirpated (McGowan, Traylor-Holzer, and Leus 2017). To be successful in this goal, captive populations must be carefully managed to maintain genetic diversity, which can be a challenge given that they are typically constrained to a small number of breeding individuals (Ralls and Ballou 1986). When populations are small, inbreeding can lead to homozygous deleterious genetic traits increasing in frequency, negatively impacting the health of offspring (Boakes, Wang, and Amos 2007). To manage this risk, managers of captive breeding programs monitor the kin relationships among individuals using either pedigrees that map families over generations, or genetic tests to estimate the relatedness among individuals. Decisions about which males and females should be bred together are then guided by the principle of minimizing relatedness or kinship between pairs (Ballou and Lacy 1995). Methods to identify suitable pairs for breeding are well developed (Ivy and Lacy 2012) and integrated into software commonly used by pro-

gram managers (e.g., PMx (Lacy, Ballou, and Pollak 2012)). However, many animals, such as primates and ungulates, do not breed as pairs and instead live and breed within groups. Designing captive breeding groups that reflect this behaviour is more difficult, as the theoretical frameworks are less well explored (Leus, Traylor-Holzer, and Lacy 2011). The design space for potential configurations of group sizes and individuals' allocations to groups can be extremely large, especially when the ideal group size is also a variable to be optimized.

In this paper we demonstrate constraint programming (CP) as a flexible framework to capture the constraints on viable designs, and cost functions for preferable designs. Despite advances in CP techniques and solver capabilities, finding feasible and optimal solutions within reasonable time frames remains a critical challenge in this domain. This paper reflects the collaboration between domain experts in the genetics of captive populations, and model developers. The outputs of the solver have already been put into practice in a Galápagos giant tortoise captive breeding program.

This paper makes the following key contributions.

- We present the first use of constraint programming (CP) to the design of captive breeding programs.
- We apply our approach to two case studies, for Galápagos giant tortoise and canid breeding systems.
- We undertake a robust performance analysis, demonstrating the scalability of the tool to meet the needs of the largest potential captive breeding programs.
- We make available an open source replication package to support the uptake of our approach in other programs.

This paper is organised as follows. In Section 2 we introduce existing approaches to captive breeding group design. Section 3 introduces the two case studies underpinning this work. Section 4 provides our constraint programming formalisation of captive breeding programs. Section 5 introduces model implementation and our experimental setup and our choice of evaluation metrics are motivated in Section 6. Section 7 presents experimental results across tortoise and canid case studies. We discuss broader implications and make our conclusions in Section 8.

2 Related Work

Existing Breeding Group Allocation Tools To determine breeding arrangements that maximise genetic diversity and minimize inbreeding, pairwise relatedness must be known between individuals in the population. Pairwise relatedness values can either be estimated from genetic markers (e.g. (Queller and Goodnight 1989) and (Wang 2002)) or inferred from an existing pedigree (Speed and Balding 2015). To our knowledge, there are two existing tools that construct breeding groups from pairwise relatedness values: *Swinger* (Sandoval-Castillo et al. 2017) and *nprcgenomekeeper* (Vinson and Raboin 2015).

Swinger accepts user defined maximum thresholds of internal relatedness, pairwise relatedness of non-breeding pairs (females:females and males:males) and average relatedness within breeding groups and across overall solutions that cannot be exceeded in possible solutions. *Swinger*

then tunes these parameters up or down accordingly until at least one, but fewer than four, best solutions are found. The runtime and computational intensity required to reach an optimal solution is therefore dependant on the user defined maximum thresholds and how much tuning is required.

Similarly, *nprcgenomekeeper* relies on thresholds to prune possible solutions, however this is not set by the user and instead set at less than half-cousins for breeding males and breeding pairs only (ignoring female:female relatedness). Individuals are then randomly assigned to groups under these thresholds. This step is repeated 10,000 times and the solution with the largest minimum group size is chosen. This minimizes relatedness among breeders but is not guaranteed to identify the optimal solution.

Matching Problems Several classes of matching problem have received significant attention in the operational research and artificial intelligence literature (Ibaraki and Katoh 1988). Of greatest relevance to this work, are efforts to model popular matching (Chisca et al. 2016) and stable matching (Sun et al. 2024) problems. Unlike related meta-heuristic approaches, we sought an approach which would converge towards optimal solutions (Olivier, Lodi, and Pesant 2018). We seek an expressive approach to capture our more complex constraints, and tractable at our problem size.

3 Case Studies and Datasets

This work is underpinned by extensive interdisciplinary collaboration to understand the needs of live captive breeding programs, to jointly formulate these requirements under our constraint programming (CP) framework, and to evaluate their impacts. We present the two case studies motivating this work, namely the conservation of the Galápagos giant tortoises, and coyote admixed with red wolf ancestry. We further outline a synthetic dataset which will be later used (§ 7) to evaluate the scalability of our approach.

Galápagos giant tortoise Galápagos giant tortoises (part of the genus *Chelonoidis*) are a group of 13 living lineages found across the archipelago and several extinct species. These populations were hugely exploited throughout the 18 and 19th centuries, with just 10% of the historical population size remaining by the 1960s when the Galápagos became protected as a national park (Tapia A. et al. 2021). Captive breeding has been very successful at restoring some populations, for example, growing the Española species from just 14 surviving individuals to its current population size of over 2000 (Milinkovitch et al. 2013). Here, we focus on the unique Floreana lost lineage recovery program (Cayot and Hunter 2021), which aims to produce a genetically diverse population of tortoises that can be used to repopulate Floreana Island. A pilot breeding program was established and ran over three breeding seasons (2011 – 2015) with three males and six females in a single corral. A genetic study of the offspring revealed unequal breeding among the males and females, which was reducing genetic diversity (Miller et al. 2018). In the following phase, the program was expanded to 20 breeders, allocated to four different corrals of three males and four females using the software *SWINGER* (Quinzin et al. 2019). Recently the program

expanded again, and there are now 51 individuals (19 males and 32 females) available for breeding. As unequal breeding remains a concern (Gray et al. 2022), there is need to explore group design and test configurations, but the design space is too large for existing software to handle in reasonable amounts of time, which provided the motivation to develop the CP approach presented here. For the analyses presented herein, pairwise relatedness was estimated between the 51 individuals from whole genome sequences using `ngsRelateV2` (Hanghøj et al. 2019).

Canids with red wolf ghost alleles Red wolves are endangered canids in North America, that became extinct in the wild and have been sustained through a captive breeding program and a small reintroduced wild population. Genomic studies have revealed red wolf ancestry in wild coyotes living in Texas and Louisiana (Murphy et al. 2019; Heppenheimer et al. 2020). Importantly, these coyotes carry both known red wolf alleles and putative ancestral red wolf genetic variation that is not present in the captive breeding program, termed “ghost alleles” (Heppenheimer et al. 2018). Thus, there is the potential to eventually increase the genetic diversity of the red wolf population by breeding in ghost alleles from the coyotes (VonHoldt et al. 2022). If such a goal is to be achieved, then breeding would need to be carefully planned among coyotes to maximize the number of ghost alleles, while avoiding inbreeding. Here we extend our CP approach to multi-objective optimization. The data for this case is pairwise relatedness of 14 coyotes estimated from whole genome sequences using PLINK (Chang et al. 2015).

Synthetic Dataset Using the allele frequencies from the contemporary population of Pinzón tortoises from (Jensen et al. 2018) we simulated 1,000 individuals, comprising 50 pairs of full siblings, 50 pairs of half siblings, and 900 unrelated individuals. We then estimated pairwise relatedness (Queller and Goodnight 1989) among them using the R package `related` in R version 4.1.1 (Pew et al. 2015; R Core Team 2020). Sexes were assigned at random at a ratio of 1:2 favouring females reflecting that organisms bred in groups (instead of pairs) typically have a polygamous mating system with more females in each group than males.

The theoretical example given of 1000 simulated individuals is an exaggeration of captive breeding populations, which generally range from tens of individuals to ~ 400 individuals (Earnhardt, Thompson, and Marhevsky 2001).

4 Captive Breeding Program Design

We define our Constraint Optimisation Problem (COP) as a 4-tuple (X, D, C, f) . $X = x_1, \dots, x_n$ is a set of variables belonging to domains $D = D_{x_1}, \dots, D_{x_n}$ respectively. The permitted values which variables may hold is captured by constraints $C = c_1, \dots, c_e$, in doing so, defining what constitutes a valid solution to the problem. Optimisation function f specifies how these permitted values should be optimised to maximise utility. The solving of COPs involves the discovery of values of X which do not violate any constraints C , whilst maximising the utility value f .

We model a captive breeding program comprising $Co \in [1, n]$ breeding corrals and a total of $I \in [1, m]$ individuals.

Individuals Individuals carry an identifier n_i , and boolean variables $s_j^{male} \in \{0, 1\}$ and $s_j^{female} \in \{0, 1\}$ if individual j is male or female, respectively. A further boolean convenience variable $d_{jk} = s_j \Leftrightarrow s_k$ indicates where individuals j and k are of opposing sex.

Individuals may belong to a set S_{prio} , representing priority individuals which must participate in the breeding program. This is useful for individuals known to be very good breeders or have other desirable characteristics.

Breeding Corrals A *corral* represents a grouping of collocated individuals that would be free to intermix and breed. A corral has a minimum size of two (representing breeding pairs) or more (in the case of group breeding regimes). Corrals are bound by a maximum capacity determined by the physical size of the corral and the requirements of animals.

We take $g_i^j \in \{0, 1\}$ as a boolean variable indicating when animal j is placed in corral i . Constraint 1 states that priority individuals must be allocated to exactly one corral for a solution to be valid, and that for non-priority individuals they may or may not be allocated to a corral.

$$\sum_{i=1}^n \begin{cases} g_i^j = 1, & \text{if } j \in S_{prio}, \forall i \in Co, j \in I \\ g_i^j \leq 1, & \text{otherwise} \end{cases}, \forall i \in Co, j \in I \quad (1)$$

For two individuals j and k , let binary variable $p_i^{jk} \in \{0, 1\}$ equal $g_i^j g_i^k$, indicating when both j and k are allocated to corral i .

Each corral has a minimum occupancy a_t^{min} and a maximum capacity a_t^{max} (Constraint 2). Each corral may include between a_m^{min} and a_m^{max} males (Constraint 3). Each corral may include between a_f^{min} and a_f^{max} females (Constraint 4). It follows that corral minimum and maximum occupancy are configured such that $a_t^{min} \geq a_m^{min} + a_f^{min}$ and $a_t^{max} \leq a_m^{max} + a_f^{max}$.

$$a_t^{min} \leq \sum_{j=1}^m g_i^j \leq a_t^{max}, \forall i \in Co \quad (2)$$

$$a_m^{min} \leq \sum_{j=1}^m s_j^{male} g_i^j \leq a_m^{max}, \forall i \in Co \quad (3)$$

$$a_f^{min} \leq \sum_{j=1}^m s_j^{female} g_i^j \leq a_f^{max}, \forall i \in Co \quad (4)$$

Each corral may accommodate up to $a_t^{nonprio}$ non-priority individuals per Constraint 5. We bound the number of non-compulsory individuals allocated to each corral, with the following corral-level constraint. These thresholds may be set flexibly at the corral level, giving practitioners flexibility as to which corrals will accept these individuals. Constraints 1 and 5 work in conjunction to ensure no non-priority individuals are allocated to a corral at the expense of a priority individual participating in the program.

$$\sum_{j=1}^m g_i^j \leq a_i^{nonprio}, \forall i \in Co, j \notin S_{prio} \quad (5)$$

Our model can easily incorporate historical breeding pair behaviour to impose mandatory ($G_m = \langle j, k \rangle \mid j, k \in [1, m], j \neq k$) and prohibited pairs (e.g. male aggression) ($G_p = \langle j, k \rangle \mid j, k \in [1, m], j \neq k$), as follows:

$$\sum_{i=1}^n \begin{cases} p_i^{jk} = 1, & \text{if } \langle j, k \rangle \in G_m, \forall i \in Co, j, k \in I \\ p_i^{jk} = 0, & \text{if } \langle j, k \rangle \in G_p \end{cases} \quad (6)$$

4.1 Objective Functions

Galápagos giant tortoise Objective Function Our objective is to minimise the sum of the pairwise relatedness of individuals allocated to the same breeding corral. Relatedness measures between each pair of individuals i and j are captured in matrix R^{ij} . Obj. Function 7 weights the pairwise relatedness of all individuals within the corrals. We square the R^{jk} term to penalise higher relatedness values. Meanwhile, Obj. Function 8 includes additional term d^{jk} so only male-female pairings contribute to the optimisation function. The flexibility to consider all pairs, or only male-female pairs, helps to tailor the function to the ecology of the organisms. For example, primates often live in matrilineal kinship groups, and so allowing higher relatedness among female-female pairs within a group can help accommodate this behaviour. Obj. Function 7 is used throughout the remainder of this work.

$$Min_a \sum_{i=1}^n \sum_{j=1}^{m-1} \sum_{k=j+1}^m (R^{jk})^2 p_i^{jk} \quad (7)$$

$$Min_a \sum_{i=1}^n \sum_{j=1}^{m-1} \sum_{k=j+1}^m d^{jk} (R^{jk})^2 p_i^{jk} \quad (8)$$

Canid Objective Function We express a multi-objective optimisation function to simultaneously minimise inbreeding (Obj. Function 6) while maximising ghost alleles (Obj. Function 8), where $ghost_j$ is an additional property defining the ghost allele value for each individual.

$$Max \sum_{i=1}^n \sum_{j=1}^m g_i^j ghost_j \quad (9)$$

Our two optimisation functions are of different magnitudes so we parameterise our function with maximum and minimum values obtained from optimisation for each function individually. Weighted Sum Method (WS) scalarisation is applied alongside normalisation of each objective function. Each objective function is normalised by independently running the solver to maximise and minimise each objective function in isolation. This provides the feasible interval for each objective function to support scaling. In situations where problems are sufficiently large that each objective cannot be solved optimally in reasonable time, approximate limits from feasible solutions may be used.

4.2 Configurable Constraints

We offer two configurable constraints allowing practitioners to refine the design goals for their breeding programs.

Minimum Assigned Individuals Our objective functions minimise the sum of pairwise relatedness values of individuals collocated within a corral. Therefore, optimal solutions are likely to allocate the minimum permitted individuals to the corrals, whilst fulfilling all other constraints. However, in captive breeding programs, it is desirable to maximise the number of individuals which take part breeding and only exclude individuals if they truly cannot be placed without violating other constraints. It is difficult to provide a natural interpretation to the relative weights of pairwise relatedness and number of individuals included in the program. This limits our ability to formalise this as a multi-objective optimisation problem. Instead we introduce a flexible constraint which specifies the minimum number of individuals z allocated to the program (Constraint 10).

$$\sum_{i=1}^n \sum_{j=1}^m g_i^j \geq z \quad (10)$$

Maximum Pairwise Relatedness We make available a global constraint which restricts the maximum permitted pairwise relatedness between any two individuals collocated in a corral. Constraint 11 ensures that no two individuals are allocated if their pairwise relatedness exceeds maximum r .

$$\sum_{i=1}^n \sum_{j=1}^{m-1} \sum_{k=j+1}^m (R^{jk}) p_i^{jk} \leq r \quad (11)$$

The choice of maximum pairwise relatedness requires careful selection. Setting a threshold too low may render a design infeasible or exclude breeders who could otherwise be included in the program. Without establishing a threshold, solutions may include pairs of individuals that are more related than would be considered acceptable in practice.

4.3 Breaking Symmetry

Symmetry in constraint programming (Gent, Petrie, and Puget 2006) concerns potential solutions which are equivalent. We apply a heuristic to break symmetry to significantly reduce the possible solution space by allocating a single individual to a corral. In turn, this prunes any potential solution which does not include this individual. In our context, the most parsimonious approach is to allocate the highest-value breeding individual to a corral. We select the individual with the lowest median pairwise relatedness across the full cohort. This follows the intuition that desirable solutions maximise genetic diversity; therefore, it is highly likely that the individual would be selected for inclusion within a corral.

5 Model Implementation

Implementation: We implement our model using the CP-SAT solver (Version 9.5.2237) made available in Google's OR-Tools optimisation suite, using the Python wrapper. Our choice of solver was guided by favourable performance demonstrated by the CP-SAT solver in MiniZinc competitions (Stuckey et al. 2014). Furthermore, CP-SAT is open source, so in our context has the potential for broader uptake and impact than a solution based on commercial solvers.

Replication Package: To ensure the adoption and broader community benefit of our work, we make available a replication package, licensed under CC-BY 4.0. We provide; a) all code to reproduce the findings of this study, b) datasets underpinning the experiments within the paper, c) a practitioners’ annex to support the application of the tool across other systems. We provide a reproducible execution environment using Docker, and instructions to run our tool in AWS.

Hardware: We seek to demonstrate the realistic performance the designer of a captive breeding program could achieve on commodity hardware. To this end, we demonstrate the performance for our solver running on a 2021 Apple MacBook Pro with M1 Pro CPU and 32GB RAM running macOS Ventura 13.6.1. Furthermore, we recognise that the behaviour of the CP-SAT solver varies to maximise the benefit of higher core counts; we therefore perform a comparison using a high-core count AWS `c6i.16xlarge` instance with 64 vCPUs featuring a Intel Xeon Platinum 8488C with an all-core turbo frequency of 3.5 GHz and 128GiB of memory.

Experimental Design: Experimental results are obtained from ten experimental runs from each configuration. We adopt best practices to minimise the impact of performance variability on cloud services (Duplyakin et al. 2023), including randomised multiple interleaved trials (Abedi and Brecht 2017). We minimise the environmental impact of our experiments by choosing sustainably-powered cloud availability zones (Henderson et al. 2020; Dodge et al. 2022).

6 Evaluation Metrics

Distribution of Pairwise Relatedness: We evaluate the quality of solutions based on the mean and maximum pairwise relatedness of individuals allocated to the same corrals.

Solution Optimality and Runtime: For a given run of the solver, t_{opt} is the time taken to obtain an optimal result. We bound the runtime of each solver run to two hours, so $t_{opt} = 7,200s$ represents a solver run which did not provide an optimal result. For constraint programming, where the majority of computation effort is required to prove optimality, we are interested to understand the time taken to obtain the first viable solution t_{first} and solutions within $x\%$ of an optimal value. Therefore, t_{99} represents the time taken to get a solution which is within 1% of the optimal obtained value.

Concordance of Solutions: We evaluate the concordance of candidate solutions to a) track differences over successive solutions from the same solver, and b) provides a measure of similarity for solutions provided by the solver under different configurations. This allows a practitioner to understand the sensitivity of solutions to their modelling assumptions and design choices. We measure similarity of solutions using the Adjusted Rand index (Vinh, Epps, and Bailey 2009; Rand 1971; Wagner and Wagner 2007) as follows:

$$\frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}} \quad (12)$$

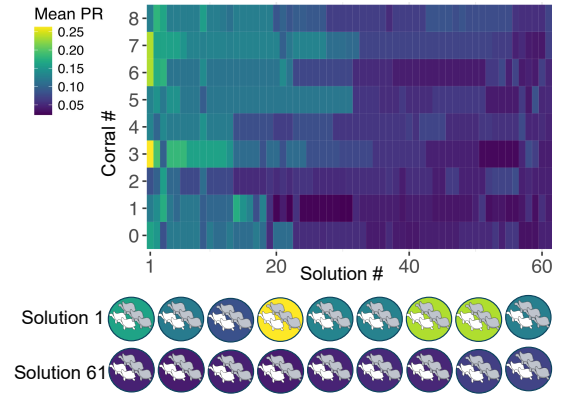


Figure 2: Heatmap of mean pairwise relatedness (PR) within the nine corrals across successive solutions for the 51 tortoises with illustrations of the corrals at solution 1 and 61.

where n_{ij} signifies the number of individuals allocate to corral i in solution A, and corral j in solution B. Meanwhile, a_i is the sum of individuals allocated in corral i of solution A which is allocated to any corral in solution B. Similarly, b_j is the sum of individuals allocated to corral j in solution B, which is allocated anywhere in Solution A.

The adjusted Rand metric has an expected value of zero for independent allocations, and 1 for identical allocations. It is a robust measure of the similarity of two solution’s pairings, and is able to compare solutions with different configurations (e.g. corral counts and sizes). We calculate a rand index for solution n of a solver run as $\{ \min(ARI(s_n, s_i)) | i \in s_0, \dots, s_{n-1} \}$ to measure the distinctiveness of the solution n to the most similar solution previously found.

7 Results

Here we present the baseline behaviour of our constraint programming approach to captive breeding program design. We then demonstrate that our approach can leverage increased parallelism, and the performance benefits for larger problem sizes. We demonstrate that through careful tuning of maximum pairwise relatedness we can dramatically increase the speed at which optimal solutions can be identified. We then show the trade-offs between maximising participation of breeders and the quality of the breeding program. Finally, we demonstrate generalisability of our approach to another system, and support for multi-objective optimisation.

7.1 Baseline Solver Behaviour

The solver uses a series of heuristics and search strategies to progressively identify feasible and improved solutions. The solver executes until a provably optimal solution is found. Execution of the solver terminates when a provably-optimal solution to the problem is found. Figure 2 illustrates the mean relatedness with corrals decrease with successive solutions.

Figure 3 illustrates the typical progress made by the solver. The left plot shows the optimality and time for each solution identified by the solver. During the first 10 seconds of runtime we see the solver produce 21 solutions, whose

	MacBook Pro				c6i.16xlarge			
	p_{first}	p_{80}	p_{99}	p_{opt}	p_{first}	p_{80}	p_{99}	p_{opt}
Galápagos Tortoise								
$c = 20$	<1s	<1s	<1s	1.33s	<1s	<1s	<1s	1.83s
$c = 30$	<1s	1.32s	4.57s	7,200s	<1s	<1s	3.12s	3,216.88s
$c = 40$	<1s	3.16s	954.31s	7,200s	<1s	1.61s	18.19	5,982.42s
$c = 51$	<1s	5.09s	1,209.12s	7,200s	3.28s	5.98s	32.17s	7,200s
Canids								
$c = 10$	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s
Synthetic								
$c = 1000$	56.92s	948.8s	5,198.11s	7,200s	41.81s	712.82s	3,918.76s	7,200s

Table 1: Solver runtimes for Galápagos Tortoise, Canid and Synthetic datasets. All reported values are median values across ten independent runs. p_{opt} equalling 7,200s represent runs which did not complete by the end of a two-hour runtime cutoff.

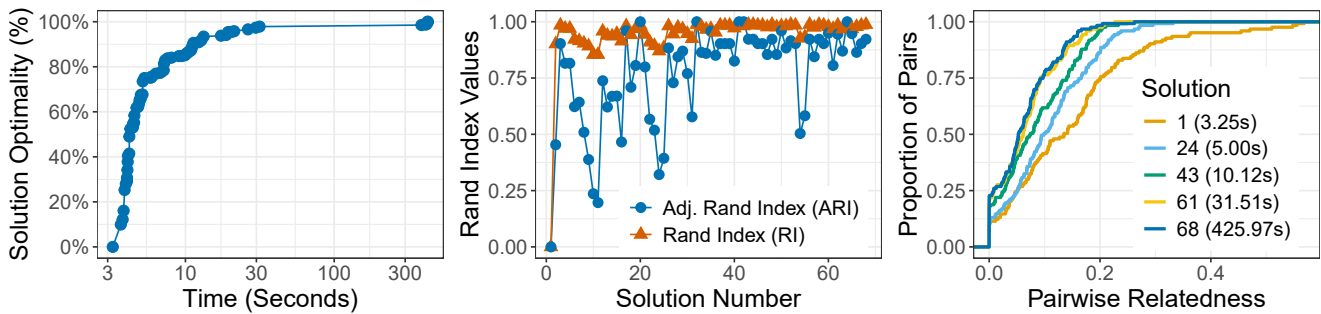


Figure 3: (Left) Optimality of solutions over time. (Center) Rand and Adjusted Rand Index values showing the concordance of successive solutions. (Right) Empirical CDF showing the distribution of pairwise relatedness within solutions.

quality rapidly converges to within 8% of the optimal value. Over the remainder of the runtime we see increased inter-solution times, before an optimal solution is identified at 59s. The centre plot demonstrates the (Adjusted) Rand index values for each solution. We see incremental improvements in the initial ~ 15 solutions. Solutions 20, 22 and 27 represent larger deviations from previously-seen solutions, and between solutions 29-35 we see modest adjustments until the provably-optimal solution is identified. The rightmost plot shows an ECDF of the pairwise relatedness values for the 1st, 10th, 20th and 35th (optimal) solution. We see a large initial improvement in relatedness values between solutions 1 and 10, while further refinement centres around the right tail, improving larger relatedness values.

7.2 Impact of Parallelism

CP-SAT (Perron, Didier, and Gay 2023) takes a portfolio approach to optimisation, leveraging ensembles of approaches - such as Local Search (Luteberget and Sartor 2023) and Large Neighbourhood Search (Shaw 1998; Danna, Rothberg, and Pape 2005) - to solve problems. CP-SAT responds well to larger numbers of workers, with hardware equipped with larger core and thread counts allowing a greater diversity of solving approaches. We demonstrate our solver’s scalability using a 64-core AWS `c6i.16xlarge` instance.

Table 1 shows that high-quality solutions can be obtained in reasonable time on commodity hardware, and increased core count not only improves our ability to identify optimal solutions within our 7,200s cutoff, but also allows near-optimal solutions to be identified more quickly. The solver was highly memory efficient (2,894MiB peak for $|c| = 51$).

7.3 Impact of Maximum Pairwise Relatedness

Judicious choice of Maximum Pairwise Relatedness (MPR) can rapidly produce high quality and provably optimal solutions. Specifying an MPR threshold reduces the potential search space to areas likely to deliver favourable solutions. Since the choice of threshold is dependent on our distribution of PR values R^{jk} , we experiment using threshold percentile $P_i(R^{jk})$ for percentile levels $i = 40, \dots, 50$.

For a configuration of 51 individuals across nine corrals of two males and three females, p_{40}, \dots, p_{45} are quickly ($\leq 0.5s$) identified as infeasible. At the inflection point between infeasible and feasible solutions (p_{46}) we see an increased runtime of 13s to identify that a solution is infeasible. Figure 4 shows results for Full (representing a two-hour solver run) and percentile values with feasible solutions. For these we see highly favourable runtimes as follows; $p_{47} = 3.33s$, $p_{48} = 3.49s$, $p_{49} = 3.52s$. Runtimes increasing beyond this point due to increased state size.

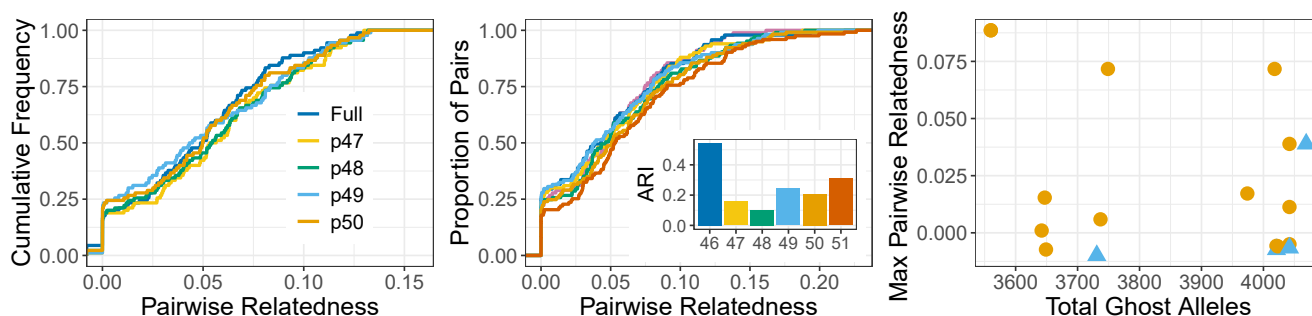


Figure 4: (Left) ECDF of pairwise relatedness (PR) for varying maximum PR thresholds. (Center) ECDF of PR of colocated individuals, for differing population size. The inset plot shows ARI values for each solution relative to the previous. (Right) Multi-objective optimisation for the canid population. The desirable region (bottom right) demonstrates the Pareto frontier between minimum PR and maximum number of ghost alleles, with Pareto optimal values highlighted as blue triangles.

In practice it is difficult to establish *a priori* the best MPR threshold to balance runtime and solution quality. This depends not only on the distribution of pairwise relatedness, but also on constraints on the breeding program design such as corral specification. For critically-endangered species with few individuals, *MPR* may be set higher to prioritise the best overall groupings. Otherwise, *MPR* should be set to cousins (0.125), then adjusted. A benefit of the CP approach is that it rapidly determines if a given threshold value is infeasible. We recommend practitioners begin with aggressive values and relax this until solutions are identified.

7.4 Impact of Maximising Population Size

There is a strong imperative to maximise the number of individuals participating in captive breeding programs. Furthermore, individuals who are excluded from the program are likely to be highly interrelated so it is not feasible for them to be kept together in a single overflow corral.

Figure 4 (Center) demonstrates the impact of increasing the number of participating individuals. Beginning with 45 individuals (9 corrals of 2 males and 3 females) we see average PR remaining consistent (with the exception of the 51 case) but more pronounced increases in both the minimum (left tail) and maximum (right tail) values. The inset figure includes ARI values, which demonstrate a marked disruption to the allocation to include the 46th individual. Two further individuals (47, 48) can be accommodated with minimal disruption to that revised allocation. Further moderate disruption is observed when placing three final individuals.

7.5 Multi-Objective Optimisation

In our case of canids with red wolf ghost alleles, we demonstrate support for multi-objective optimisation to allocate coyotes to maximize the number of ghost alleles, while avoiding inbreeding. We weight cost functions 7 and 9 with $\alpha = 0, \dots, 1$ and $(1 - \alpha)$ respectively, in steps of 0.05.

Figure 4 (Right) presents the Pareto frontier of solutions, with the bottom right representing the most favourable solutions. The total runtime for single-objective optimisation, plus the parameter sweep of weighting values was ≤ 8 s.

8 Conclusions and Broader Impact

We have demonstrated the applicability of CP for optimisation of group-living captive breeding programs. We show the flexibility of our approach through two real-world conservation case studies. In the case of the Galápagos giant tortoise Floreana lost lineage recovery program, updated breeding groups built using this tool have already been adopted by the Galápagos National Park and the breeding season underway. For the canid case, this approach will be key for modelling scenarios to prove whether it would be feasible to selectively breed the coyotes to the extent that the ghost alleles could be introduced into the red wolf population.

A core value of this work was to ensure the approach could be easily used on commodity hardware, given breeding program managers may not have budget or access to large-scale compute. Our approach makes this possible, offering a significant leap forward in scalability, solving scenarios up to 1,000 individuals rapidly. The only input required is pairwise relatedness data which can be readily produced from a pedigree or genetic study of any species. Our model can readily adapt to changing circumstances (e.g. deaths, additions, evolving behavioural compatibility). Our highly favourable runtime allows these impacts to be evaluated rapidly. Further work will extend support for multi-objective optimisation for non-convex Pareto frontiers (e.g. weighted Chebyshev scalarisation) and participatory co-design of an accessible user interface with stakeholders.

This tool is a helper in the decision-making process of designing captive breeding programs, and allows human intuition to make the final management decision, for example, the considering the other needs of captive breeding outside of relatedness, such as animal behaviour, husbandry best practices and financial constraints. Breeding decisions that are informed using this tool will become part of the genetic legacy of the captive population and will play a key part in endangered species recovery. This work aligns with the United Nations Sustainable Development Goals (SDG) to preserve life on land (SDG 15) and below water (SDG 14), and the Convention on Biological Diversity Target 4, to halt species extinction and protect genetic diversity (CBD 2022).

References

- Abedi, A.; and Brecht, T. 2017. Conducting repeatable experiments in highly variable cloud computing environments. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering*, 287–292.
- Ballou, J. D.; and Lacy, R. C. 1995. Identifying genetically important individuals for management of genetic diversity in pedigreed populations. *Population management for survival and recovery*.
- Boakes, E.; Wang, J.; and Amos, W. 2007. An investigation of inbreeding depression and purging in captive pedigreed populations. *Heredity*, 98(3): 172–182.
- Cayot, L. J.; and Hunter, E. A. 2021. Floreana and Pinta Islands: Restoring tortoise populations through lost lineage recovery. In *Galapagos Giant Tortoises*, 465–481. Elsevier.
- Chang, C. C.; Chow, C. C.; Tellier, L. C.; Vattikuti, S.; Purcell, S. M.; and Lee, J. J. 2015. Second-generation PLINK: rising to the challenge of larger and richer datasets. *Giga-science*, 4(1): s13742–015.
- Chisca, D.; Siala, M.; Simonin, G.; and O’Sullivan, B. 2016. A CP-Based Approach to Popular Matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Danna, E.; Rothberg, E. E.; and Pape, C. L. 2005. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102: 71–90.
- Dodge, J.; Prewitt, T.; Tachet des Combes, R.; Odmark, E.; Schwartz, R.; Strubell, E.; Luccioni, A. S.; Smith, N. A.; DeCario, N.; and Buchanan, W. 2022. Measuring the Carbon Intensity of AI in Cloud Instances. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’22, 1877–1894. New York, NY, USA.
- Duplyakin, D.; Ramesh, N.; Imburgia, C.; Al Sheikh, H. F.; Jain, S.; Tekta, P.; Maricq, A.; Wong, G.; and Ricci, R. 2023. Avoiding the Ordering Trap in Systems Performance Measurement. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, 373–386.
- Earnhardt, J. M.; Thompson, S. D.; and Marhevsky, E. A. 2001. Interactions of target population size, population parameters, and program management on viability of captive populations. *Zoo Biology*, 20(3): 169–183.
- Gent, I. P.; Petrie, K. E.; and Puget, J.-F. 2006. Symmetry in constraint programming. *Foundations of Artificial Intelligence*, 2: 329–376.
- Gray, R.; Fusco, N.; Miller, J. M.; Tapia, W.; Mariani, C.; Caccone, A.; and Jensen, E. L. 2022. Temporal monitoring of the Floreana Island Galapagos giant tortoise captive breeding program. *Integrative and Comparative Biology*, 62(6): 1864–1871.
- Hanghøj, K.; Moltke, I.; Andersen, P. A.; Manica, A.; and Korneliussen, T. S. 2019. Fast and accurate relatedness estimation from high-throughput sequencing data in the presence of inbreeding. *GigaScience*, 8(5): giz034.
- Henderson, P.; Hu, J.; Romoff, J.; Brunskill, E.; Jurafsky, D.; and Pineau, J. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248): 1–43.
- Heppenheimer, E.; Brzeski, K. E.; Hinton, J. W.; Chamberlain, M. J.; Robinson, J.; Wayne, R. K.; and vonHoldt, B. M. 2020. A genome-wide perspective on the persistence of red wolf ancestry in southeastern canids. *Journal of Heredity*, 111(3): 277–286.
- Heppenheimer, E.; Brzeski, K. E.; Wooten, R.; Waddell, W.; Rutledge, L. Y.; Chamberlain, M. J.; Stahler, D. R.; Hinton, J. W.; and VonHoldt, B. M. 2018. Rediscovery of red wolf ghost alleles in a canid population along the American Gulf Coast. *Genes*, 9(12): 618.
- Ibaraki, T.; and Katoh, N. 1988. *Resource allocation problems: algorithmic approaches*. MIT press.
- Ivy, J. A.; and Lacy, R. C. 2012. A comparison of strategies for selecting breeding pairs to maximize genetic diversity retention in managed populations. *Journal of Heredity*, 103(2): 186–196.
- Jensen, E. L.; Edwards, D. L.; Garrick, R. C.; Miller, J. M.; Gibbs, J. P.; Cayot, L. J.; Tapia, W.; Caccone, A.; and Russello, M. A. 2018. Population genomics through time provides insights into the consequences of decline and rapid demographic recovery through head-starting in a Galapagos giant tortoise. *Evolutionary Applications*, 11(10): 1811–1821.
- Lacy, R. C.; Ballou, J. D.; and Pollak, J. P. 2012. PMx: software package for demographic and genetic analysis and management of pedigreed populations. *Methods in Ecology & Evolution*, 3(2).
- Leus, K.; Traylor-Holzer, K.; and Lacy, R. 2011. Genetic and demographic population management in zoos and aquariums: recent developments, future challenges and opportunities for scientific research. *International Zoo Yearbook*, 45(1): 213–225.
- Luteberget, B.; and Sartor, G. 2023. Feasibility Jump: an LP-free Lagrangian MIP heuristic. *Mathematical Programming Computation*, 15(2): 365–388.
- McGowan, P. J.; Traylor-Holzer, K.; and Leus, K. 2017. IUCN guidelines for determining when and how ex situ management should be used in species conservation. *Conservation Letters*, 10(3): 361–366.
- Milinkovitch, M. C.; Kanitz, R.; Tiedemann, R.; Tapia, W.; Llerena, F.; Caccone, A.; Gibbs, J. P.; and Powell, J. R. 2013. Recovery of a nearly extinct Galápagos tortoise despite minimal genetic variation. *Evolutionary applications*, 6(2): 377–383.
- Miller, J. M.; Quinzin, M. C.; Scheibe, E. H.; Ciofi, C.; Villalva, F.; Tapia, W.; and Caccone, A. 2018. Genetic pedigree analysis of the pilot breeding program for the rediscovered Galapagos giant tortoise from Floreana Island. *Journal of Heredity*, 109(6): 620–630.
- Murphy, S. M.; Adams, J. R.; Cox, J. J.; and Waits, L. P. 2019. Substantial red wolf genetic ancestry persists in wild canids of southwestern Louisiana. *Conservation Letters*, 12(2): e12621.
- Olivier, P.; Lodi, A.; and Pesant, G. 2018. A comparison of optimization methods for multi-objective constrained bin packing problems. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*:

- 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26–29, 2018, *Proceedings 15*, 462–476. Springer.
- Perron, L.; Didier, F.; and Gay, S. 2023. The CP-SAT-LP Solver (Invited Talk). In *29th International Conference on Principles and Practice of Constraint Programming (CP 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik.
- Pew, J.; Muir, P. H.; Wang, J.; and Frasier, T. R. 2015. related: an R package for analysing pairwise relatedness from codominant molecular markers. *Molecular ecology resources*, 15(3): 557–561.
- Queller, D. C.; and Goodnight, K. F. 1989. Estimating relatedness using genetic markers. *Evolution*, 43(2): 258–275.
- Quinzin, M. C.; Sandoval-Castillo, J.; Miller, J. M.; Beheregaray, L. B.; Russello, M. A.; Hunter, E. A.; Gibbs, J. P.; Tapia, W.; Villalva, F.; and Caccone, A. 2019. Genetically informed captive breeding of hybrids of an extinct species of Galapagos tortoise. *Conservation Biology*, 33(6): 1404–1414.
- R Core Team. 2020. RA language and environment for statistical computing, R Foundation for Statistical. *Computing*.
- Ralls, K.; and Ballou, J. 1986. Captive breeding programs for populations with a small number of founders. *Trends in Ecology & Evolution*, 1(1): 19–22.
- Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336): 846–850.
- Sandoval-Castillo, J.; Attard, C. R.; Marri, S.; Brauer, C. J.; Möller, L. M.; and Beheregaray, L. B. 2017. SWINGER: A user-friendly computer program to establish captive breeding groups that minimize relatedness without pedigree information. *Molecular Ecology Resources*, 17(2): 278–287.
- Shaw, P. 1998. Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming*, 417–431. Springer.
- Speed, D.; and Balding, D. J. 2015. Relatedness in the post-genomic era: is it still useful? *Nature Reviews Genetics*, 16(1): 33–44.
- Sun, Z.; Yamada, N.; Takenami, Y.; Moriwaki, D.; and Yokoo, M. 2024. Stable Matchings in Practice: A Constraint Programming Approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 22377–22384.
- Tapia A., W.; Sevilla, C.; Málaga, J.; and Gibbs, J. P. 2021. Chapter 20 - Tortoise populations after 60 years of conservation. In Gibbs, J. P.; Cayot, L. J.; and Aguilera, W. T., eds., *Galapagos Giant Tortoises*, Biodiversity of World: Conservation from Genes to Landscapes, 401–432. Academic Press. ISBN 978-0-12-817554-5.
- Vinh, N. X.; Epps, J.; and Bailey, J. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, 1073–1080.
- Vinson, A.; and Raboin, M. J. 2015. A practical approach for designing breeding groups to maximize genetic diversity in a large colony of captive rhesus macaques (*Macaca mulatta*). *Journal of the American Association for Laboratory Animal Science*, 54(6): 700–707.
- VonHoldt, B. M.; Hinton, J. W.; Shutt, A. C.; Murphy, S. M.; Karlin, M. L.; Adams, J. R.; Waits, L. P.; and Brzeski, K. E. 2022. Reviving ghost alleles: Genetically admixed coyotes along the American Gulf Coast are critical for saving the endangered red wolf. *Science advances*, 8(26): eabn7731.
- Wagner, S.; and Wagner, D. 2007. Comparing clusterings: an overview.
- Wang, J. 2002. An estimator for pairwise relatedness using molecular markers. *Genetics*, 160(3): 1203–1215.
- Western, D. 1992. The biodiversity crisis: a challenge for biology. *Oikos*, 29–38.