Reduced Inbreeding Seed Orchard Engine: A Web-Based Application for Optimising Seed Orchard Layouts

Divya Prakash[1]*, Diwakar Ranjan[2], Ish Prakash[3], Tanya Singh[4], Naseer Mohammad[1] and Ayyanadar Arunachalam[5]

[1]G&TI Division, ICFRE-Tropical Forest Research Institute, Jabalpur, India.
[2]Dr. B.R. Ambedkar Bihar University, Muzaffarpur, Bihar, India.
[3]Dr. Rajendra Prasad Central Agricultural University, Pusa, Bihar, India.
[4]Academy of Scientific and Innovative Research, Uttar Pradesh, India.
[5]ICAR-Central Agroforestry Research Institute, Jhansi, Uttar Pradesh, India.

* Correspondence:
Divya Prakash
Email: divyap@icfre.org, divya.pr3345@gov.in
Telephone: 0761-2840241

Abstract

In advanced seed orchards, factors such as shared clone origins and uneven planting densities heighten the chances of inbreeding. The RISE tool advances beyond traditional minimum-inbreeding and quadratic assignment frameworks by incorporating comprehensive coancestry data, gap-filling for plant losses, variable clone counts, and dynamic features like distance graphs and selectable clone emphasis in an online environment. This research describes an innovative browser-accessible program that creates efficient orchard arrangements for practical forest management, employing a preliminary placement strategy paired with a streamlined annealing process to reduce the inbreeding indicator, *d_min*. The system accommodates custom grid setups, the number of clones, clonal replicates, and sibling links, producing arrangements through spreadsheet exports and engaging visual displays. Relative to arbitrary configurations, the program yields notably reduced *d_min* levels (an indicator of mating hazards). It provides statistical analysis of the resulting orchard, such as paired t-test, Kolmogorov-Smirnov test and Levene's test, validating improvements across cases. It can handle complex pedigrees and co-ancestry estimates from marker based genomic studies. The tool suits forestry practitioners, delivering an effective option for orchards of diverse scales. The program's architecture, approach, and forestry applications are explored, underscoring its role in advancing orchard oversight.

Keywords: Seed orchard design, quadratic assignment problem, simulated annealing, web application, forestry optimization

1. Introduction

Seed orchards are critical for producing genetically improved seeds in forest tree breeding programs (El-Kassaby 1995). Advanced-generation orchards often include clones with common ancestry, increasing the risk of inbreeding through selfing or mating among related clones (Hodge and White 1993). Reduced genetic gain in forest plantations and seed yield losses from empty seeds or low germination rates are the results of inbreeding in seed orchards (Nicodemus 2025).

In the early stages of forest tree breeding programs, a wide range of seed orchard layouts were proposed, including chessboard, pure rows, randomized complete blocks, fixed blocks, rotating blocks, reversed blocks, unbalanced incomplete blocks, balanced incomplete blocks, cyclic incomplete blocks, balanced lattice, permutated neighborhoods, and systematic designs (Giertych 1975). Since it served as the foundation for later software code improvements, the computerized Permutated Neighborhood technique introduced by Bell and Fletcher (1978) , called Computer Organized Orchard Layouts (COOL), was used to establish some of the most well-known seed orchards in Canada. Chakravarty and Bagchi (Chakravarty and Bagchi 1993, 1994) expanded on the COOL concept under the moniker Permutated Neighborhood Seed Orchard Design. However, relatedness among clones was not an intended feature of the original COOL program code or its upgraded version. Furthermore, despite the COOL design's exclusion zones being met, the resulting distances between genetic entities (ramets of the same clone) are not optimised due to local assignment; as a result, the crop's projected inbreeding rate is not minimised.

The minimum-inbreeding (MI) seed orchard design mitigates these risks by optimising the spatial arrangement of clones to minimise inbreeding, formulated as a quadratic assignment problem (QAP) (Lstibůrek and El-Kassaby 2010). The MI

DOI:10.2478/sg-2026-0004
edited by the Thünen Institute of Forest Genetics

approach uses a metric, *d_min*, which sums the inverse squared distances between ramets of the same clone and half-sibs, serving as a proxy for inbreeding risk under assumptions of isotropic dispersal and distance-decay mating (Lstibůrek et al. 2015). In recent years, R^2SCR Design (El-Kassaby et al. 2014), ONA design (Chaloupková et al. 2016), and IAPGA approach (Yang et al. 2020) have been developed for seed orchard layouts. When it comes to assigning ramets of related clones or ramets of the same clone with the maximum distance along the orchard's grid, the MI-design outperforms alternative methods (Chaloupková et al. 2016; Liesebach et al. 2021).

While the MI design has been expanded to large-scale orchards using advanced heuristics like genetic-tabu search (Lstibůrek et al. 2015), its operational adoption is limited by computational complexity and the need for technical expertise. Seed orchards, common in regional forestry programs, require accessible tools that balance optimization with usability. Web-based applications offer a solution, providing intuitive interfaces and rapid computation for non-expert users (Marques et al. 2021).

This study introduces the Reduced Inbreeding Seed Orchard Engine (RISE), a Flask-based web application that implements a heuristic solution to the QAP for small to large-scale seed orchards, evaluated against randomised baselines. RISE innovates beyond existing MI/QAP tools by integrating general coancestry matrices, in-filling for mortality, unbalanced ramet deployment, and interactive outputs like distance histograms and clone highlighting in a browser-based platform.

2. Materials and Methods

2.1 Application Design

The application is built using Flask, a Python web framework, and hosted on a server (https://flask-app-463764081874.asia-south1.run.app). The complete Python source code is provided as a supplementary file. It features a user-friendly interface accessible via web browsers, with input fields for orchard grid dimensions (rows and columns), number of clones (N_C), ramets per clone (N_R), and optional half-sib sets or upload of a full coancestry matrix (Supplementary Table 1) for complex relationships (e.g., full-sibs at 0.25 coancestry, genomic values as continuous). The tool supports partially stocked grids with empty cells or surplus positions, enabling in-filling of existing orchards post-mortality. RISE supports unbalanced ramets per clone and upload of existing layouts (CSV) for in-filling post-mortality (Supplementary Table 2), with optimisation only on available positions. While RISE generates layouts on rectangular grids, its in-filling feature enables users to accommodate non-plantable areas (e.g., due to terrain constraints) by placing extra clones at non-plantable areas in the uploaded existing layout CSV. Outputs include CSV layouts, Plotly visualisations (with interactive clone highlighting and relative shading), histograms of pairwise related-ramet distances, adjacency tables for fixed positions, and statistical comparisons (e.g., paired t-test, Kolmogorov-Smirnov test, and Levene's test).

2.2 Mathematical Framework

The RISE seed orchard design is formulated as a quadratic assignment problem (QAP), as proposed by Lstibůrek and El-Kassaby (2010), which assigns ramets to grid positions to minimise the inbreeding proxy metric *d_min*. This metric sums the inverse squared Euclidean distances between ramets of the same clone (selfing component) and half-sibs (weighted by 0.5 to reflect coancestry), serving as a proxy for inbreeding risk under assumptions of isotropic pollen dispersal and distance-decay mating probabilities (Lstibůrek et al. 2015). Limitations include decoupling from realised mating due to directional wind, heterogeneous fecundity, flowering asynchrony, or landscape barriers.

Uniquely, RISE extends this framework by incorporating a general coancestry matrix (θ), where *d_min* is computed as the sum over all related ramet pairs of θ_{ij} divided by the squared distance between them. Here, θ_{ij} represents the coancestry coefficient between clones i and j (e.g., 0.5 for self, 0.125 for half-sibs, 0.25 for full-sibs). Users can upload this matrix as a CSV file, enabling optimisation for complex pedigrees beyond simple half-sib sets. For marker-based studies, the relationship matrix should provide pairwise coancestry estimates, such as realised kinship from genomic markers (e.g., identity-by-state or identity-by-descent coefficients derived from SNPs or microsatellites) (VanRaden, 2008). This general θ-matrix formulation is scaled identically to the original half-sib/selfing decomposition in Lstibůrek and El-Kassaby (2010) and Lstibůrek et al. (2015), where same-clone ramet pairs contribute with $\theta = 0.5$ and half-sib ramet pairs with $\theta = 0.125$, ensuring the inbreeding proxy remains directly comparable. These continuous values (typically ranging from 0 to 0.5) allow RISE to account for empirical relatedness, improving accuracy over pedigree assumptions in advanced breeding programs.

2.3 Optimization Algorithm

Given the computational complexity of QAP, RISE employs heuristics: a greedy initialization that places ramets to maximize minimum distances to related ones (sampling up to 50 candidates for efficiency), followed by lightweight simulated annealing to refine the layout. A proposed swap is accepted if it yields $\Delta d\_min \leq 0$; otherwise it is accepted with probability $\exp(-\Delta d\_min / T)$, where T is the current temperature (initial $T = 500$, cooling rate $= 0.98$, final $T = 0.1$; 500–1000 iterations scaled by grid size). Incremental updates compute only affected *Δd_min* changes for speed. The Python code for the application is provided as a supplementary file. It should be noted that a more intensive computation could be performed by increasing the initial temperature, cooling rate, and number of iterations; preliminary tests showed diminishing returns beyond ~50 candidates, and the parameter is fully configurable in the source code. Users may approach the corresponding author for more intensive computations.

2.4 Random Layout Generation and Implementation

Random baselines are generated by shuffling ramet assignments randomly (10 replications per run) for comparison, with standard deviation and coefficient of variation derived from these. The Flask-based app uses NumPy for arrays and Plotly for visualizations; inputs include grid dimensions, clone/ramet counts (balanced or unbalanced), half-sib sets or coancestry matrix uploads, and existing layouts for in-filling.

Statistical comparisons are generated automatically by the app and include: (i) a paired t-test to compare mean *d_min* between RISE and random layouts (10 random replications per run), (ii) a Kolmogorov–Smirnov test to evaluate differences in the distribution of pairwise related-ramet distances, and (iii) Levene's test to compare variances in these distances. The app outputs the test statistics and p-values for user inspection; the present manuscript reports only descriptive summary statistics.

The RISE application was evaluated on a 20×20 grid (400 positions) using the scenarios from Chaloupková et al. (2016) to enable direct comparison with the Optimum Neighborhood Algorithm (ONA), Minimum Inbreeding (MI), R^2SCR, and randomized (RD) layouts. To assess robustness, 30 independent runs were performed for the two focus scenarios used in Chaloupková et al. (2016) for cross-method comparison:

- Balanced: 40 clones × 10 ramets each.
- Unbalanced: 10 clones × 20 ramets, 18 clones × 10 ramets, and 4 clones× 5 ramets.

No relatedness was included (relationship matrix off-diagonals zero), so *d_min* reflects only selfing avoidance via inverse squared distances between same-clone ramets.

3. Results and Discussion

3.1 Performance and Comparison Against Existing Methods

The cross-method comparison in Table 1 was performed under a selfing-only setting (all off-diagonal elements of the coancestry matrix set to zero), exactly as in Chaloupková et al. (2016). The comparison of *d_min* values for RISE in comparison to earlier designs has been presented in Table 1. The RISE layout for balanced and unbalanced scenarios has been presented in Figure 1 and Figure 2, respectively. RISE achieved mean *d_min* values of 17.24 (balanced) and 22.34 (unbalanced), representing substantial reductions compared to random layouts. These values were also markedly lower than R^2SCR, ONA and MI in both scenarios, despite MI's more intensive optimisation. The low standard deviation (0.71 and 0.43) across 30 runs indicates high consistency in RISE outputs. A representative output of the application for in-filling is given in Supplementary Figure 1. An illustrative optimisation using a non-zero coancestry matrix (including half-sib relationships among 10 clones) is provided in Supplementary Table 1 and Supplementary Figure 2.

These results demonstrate that RISE's greedy initialisation combined with lightweight simulated annealing effectively minimises the *d_min* proxy for inbreeding risk (selfing component) under assumptions of isotropic pollen dispersal and distance-decay mating probabilities (Lstibůrek et al. 2015). Limitations of the proxy include potential decoupling from realized mating due to directional wind, heterogeneous fecundity, flowering asynchrony, or landscape barriers.

RISE generated these layouts within minutes on standard hardware (Intel Core i3, 8 GB RAM), in contrast to the higher computational demands of MI and ONA approaches. This efficiency, together with the web-based interface, makes RISE particularly suitable for operational forestry where rapid, accessible design is a priority.

The quantitative benchmark (Table 1) confirms RISE's competitive performance on *d_min* relative to MI, ONA, and R^2SCR results, while offering advantages in runtime, user accessibility, support for unbalanced ramet deployment, in-filling of empty/surplus positions, and general coancestry matrix handling.

3.2 Usability and Implications for Forestry

The application enables forestry programs to optimise layouts without specialised software or supercomputing resources. By reducing inbreeding, the app enhances the genetic diversity of seed crops, improving seedling vigour and adaptation (El-Kassaby, 2003). Its scalability to large-sized grids and support for half-sib constraints make it versatile for regional orchards, where relatedness is a common concern. While RISE generates layouts on rectangular grids, its in-filling feature enables users to accommodate non-plantable areas (e.g., due to terrain constraints) by designating them as clonal sites (putting extra clones) in an uploaded existing layout CSV. This approach also allows for the optimisation of irregular orchard geometries and varying sizes, bounded by the specified grid, ensuring adaptable designs for real-world planting sites.

The app also aligns with sustainable forestry goals, as optimized seed orchards contribute to resilient forest ecosystems. Its rapid computation supports iterative design, allowing managers to test multiple configurations before planting. Compared to traditional designs like randomized, replicated, staggered clonal-row layouts (El-Kassaby et al. 2014), RISE app offers a more systematic approach to inbreeding minimization, potentially increasing seed yield and quality. Its user friendliness allows it to be used by both researchers and forest personnel with ease. The ability to handle irregular orchard geometries via in-filling (by designating non-plantable areas as fixed clonal positions in uploaded CSVs) further enhances RISE's practical utility for real-world planting sites with terrain constraints or mortality.

Table 1: Inbreeding metric (*d_min*) results from RISE compared to Chaloupková et al. (2016)

Scenario	Method	*d_min* (mean)	SD	Range (min–max)	Remarks
Balanced (40×10)	RISE	17.24	0.71	15.96–18.63	This study
Balanced (40×10)	ONA	32.25	—	—	Chaloupková et al. (2016)
Balanced (40×10)	MI	21.58	—	—	Chaloupková et al. (2016)
Balanced (40×10)	R^2SCR	40.34	—	—	Chaloupková et al. (2016)
Balanced (40×10)	RD (random)	66.30	—	—	Chaloupková et al. (2016)
Unbalanced (mixed)	RISE	22.34	0.43	21.52–23.21	This study
Unbalanced (mixed)	ONA	60.94	—	—	Chaloupková et al. (2016)
Unbalanced (mixed)	MI	51.39	—	—	Chaloupková et al. (2016)
Unbalanced (mixed)	R^2SCR	59.66	—	—	Chaloupková et al. (2016)
Unbalanced (mixed)	RD (random)	98.62	—	—	Chaloupková et al. (2016)

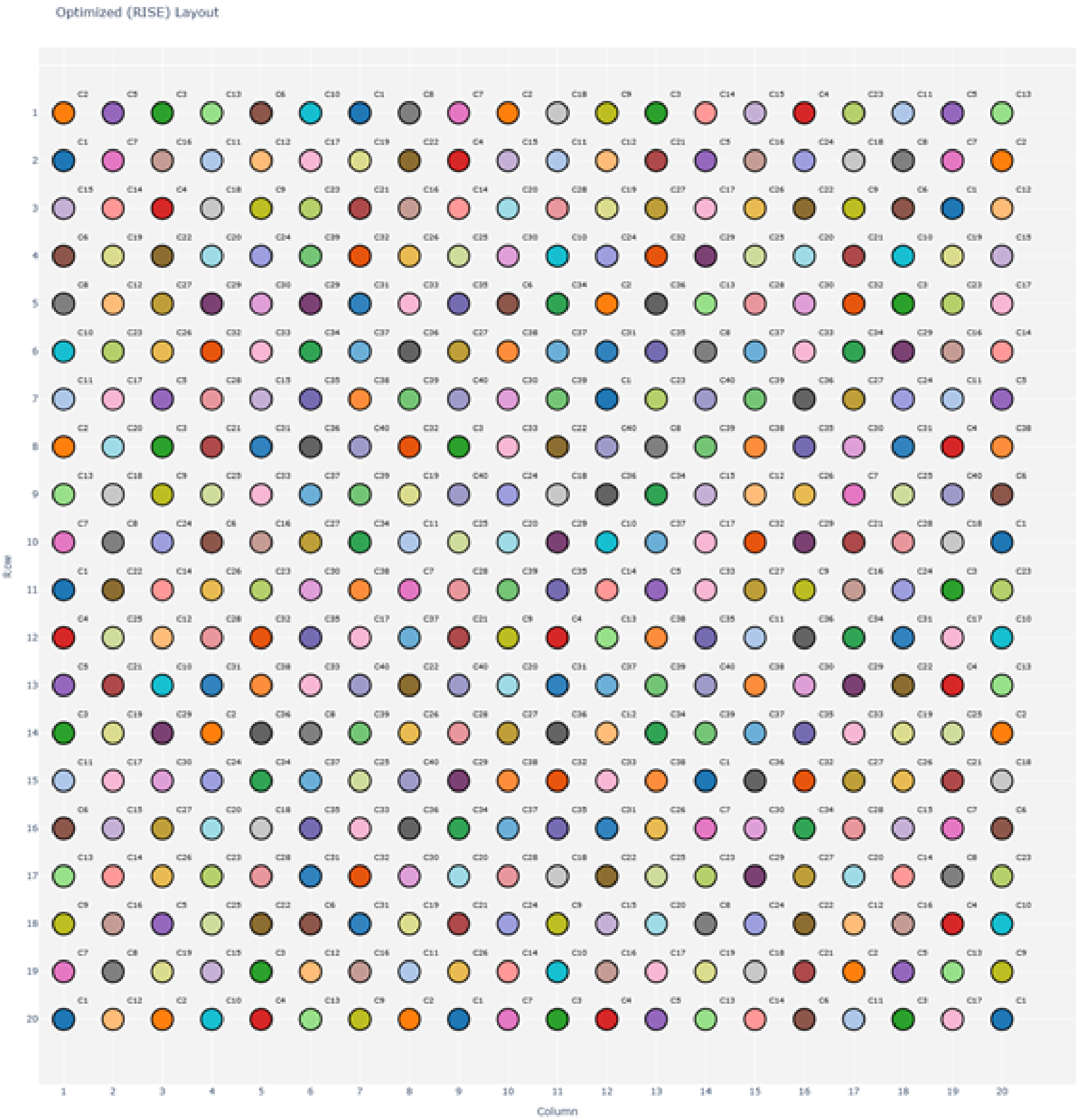

Figure 1
RISE layout of the balanced scenario

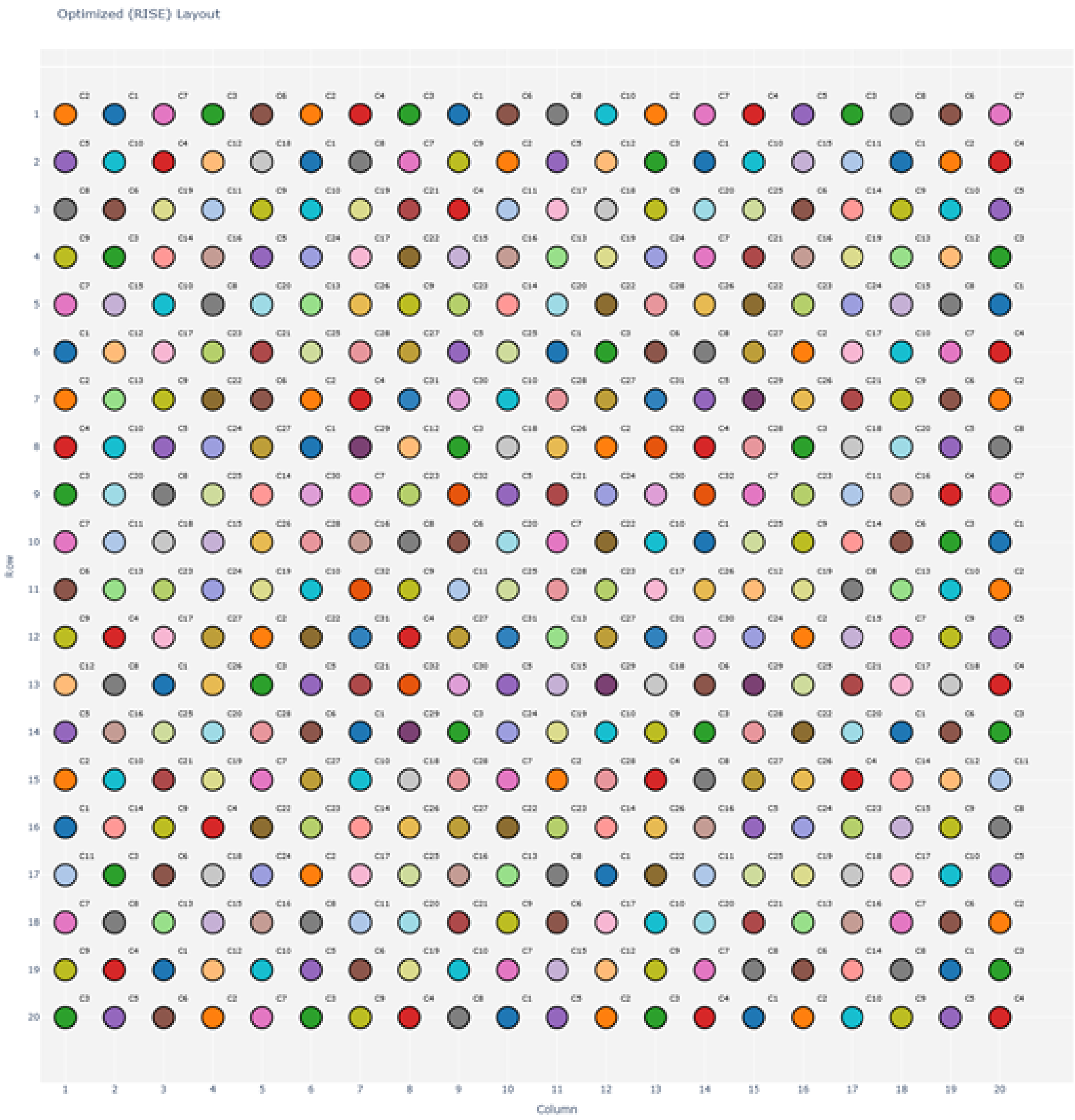

Figure 2
RISE layout of the unbalanced scenario.

4. Conclusion

This research unveils an online platform that brings minimum-inbreeding orchard planning into practice, using initial positioning and annealing techniques to curb mating among relatives. Its effectiveness, ease of use, and availability position it as an essential resource for orchards ranging from modest to extensive, meeting the demands of forestry experts in progressive breeding efforts. The application can handle complex pedigrees and estimates of co-ancestry from marker-based genomic studies. The platform's speed and functional results promote its integration into everyday forest operations.

Acknowledgements

The authors thank the open-source community for contributions to Flask, NumPy, and Plotly. The corresponding author thanks the Director, ICFRE-Tropical Forest Research Institute, for institutional support. We thank the anonymous reviewers for their comments towards the improvement of the manuscript.

References

Bell G and A Fletcher (1978) Computer organised orchard layouts (COOL) based on the permutated neighbourhood design concept. Silvae Genetica 27:223–225.

Chakravarty G and S Bagchi (1994) Short note: Enhancement of the computer program of the permutated neighbourhood seed orchard design. Silvae Genetica 43:177–178.

Chakravarty G and S Bagchi (1993) A computer program for permutated neighbourhood seed orchard design. Silvae Genetica 42:1–1.

Chaloupková K, J Stejskal, Y A El-Kassaby and M Lstibůrek (2016) Optimum neighborhood seed orchard design. Tree Genetics & Genomes 12:1–7. https://doi.org/10.1007/s11295-016-1067-y

El-Kassaby Y (1995) Evaluation of the tree-improvement delivery system: factors affecting genetic potential. Tree Physiology 15:545–545. https://doi.org/10.1093/treephys/15.7-8.545

El-Kassaby Y A, M Fayed, J Klápště and M Lstibůrek (2014) Randomized, replicated, staggered clonal-row (R^2SCR) seed orchard design. Tree Genetics & Genomes 10:555–563. https://doi.org/10.1007/s11295-014-0703-7

Giertych M (1975) Seed orchard designs, pp. 25–37. In: Seed Orchards: A Joint Production by Specialist Members of the IUFRO Working Party on Seed Orchards (S203.03). Forestry Commission Bulletin.

Hodge G R and T L White (1993) Advanced-generation wind-pollinated seed orchard design. New Forests 7:213–236. https://doi.org/10.1007/bf00127387

Liesebach H, K Liepe and C Bäucker (2021) Towards new seed orchard designs in Germany – a review. Silvae Genetica 70:84–98. https://doi.org/10.2478/sg-2021-0007

Lstibůrek M and Y A El-Kassaby (2010) Minimum-inbreeding seed orchard design. Forest Science 56:603–608. https://doi.org/10.1093/forestscience/56.6.603

Lstibůrek M, J Stejskal, A Misevicius, et al. (2015) Expansion of the minimum-inbreeding seed orchard design to operational scale. Tree Genetics & Genomes 11:1–8. https://doi.org/10.1007/s11295-015-0842-5

Marques S M, J Planas-Iglesias and J Damborsky (2021) Web-based tools for computational enzyme design. Current Opinion in Structural Biology 69:19–34. https://doi.org/10.1016/j.sbi.2021.01.010

Nicodemus A (2025) Seed production systems for forest tree species. In: Textbook of Forest Science. Springer, pp. 633–657. https://doi.org/10.1007/978-981-97-8289-5_29

VanRaden P M (2008) Efficient methods to compute genomic predictions. Journal of dairy science 91: 4414-4423. https://doi.org/10.3168/jds.2007-0980

Yang B, H Sun, J Qi, et al. (2020) Improved genetic distance-based spatial deployment can effectively minimize inbreeding in seed orchard. Forest Ecosystems 7:1–11. https://doi.org/10.1186/s40663-020-0220-0