# A Multi-objective Model Search Algorithm for Linear Regression

Anas Mifrani, Philippe Saint-Pierre, Nicolas Savy

# A Multi-objective Model Search Algorithm for Linear Regression

Anas Mifrani (Corresponding author)
Toulouse Mathematics Institute
UMR5219 - University of Toulouse; CNRS - UPS IMT
and
Philippe Saint-Pierre
Toulouse Mathematics Institute
UMR5219 - University of Toulouse; CNRS - UPS IMT
and
Nicolas Savy
Toulouse Mathematics Institute
UMR5219 - University of Toulouse; CNRS - UPS IMT

May 20, 2023

**Abstract**

Inherent in model selection is the problem of simultaneously optimizing multiple performance metrics. Some of these metrics express potentially conflicting criteria, like accuracy and simplicity. Pareto optimization is a branch of mathematical optimization that deals precisely with problems involving conflicting objective functions. In this article, an algorithm was developed that searches automatically for Pareto optimal linear regression models given a dataset and a set of performance metrics. The optimization task was framed as one of sequential variable selection on a graph. A search strategy was proposed that draws on ant colony optimization, a probabilistic technique well suited for graph-based problems. Experiments were run in which the metrics to be minimized were the root-mean-square error, expressing accuracy, and the number of coefficients, expressing simplicity. To substantiate the usefulness of our algorithm, cases were presented in which it outperformed AIC-based stepwise regression. Results suggested that our algorithm copes well with small datasets and correlated predictors, that it is efficient and that it informs model selection. Key properties of our algorithm were discussed and areas of improvement highlighted.

*Keywords:* Model selection, Multi-objective optimization, Ant colony optimization, Pareto front, Tradeoffs.

# 1 Introduction and Motivation

Part of the complexity of model selection is that it is usually motivated by multiple and potentially competing objectives. A typical example is when a model has to conform as much as possible to available data, but also has to be sufficiently simple for it to generalize and be interpretable [1, 2]. Some sort of compromise between accuracy and simplicity is often necessary because complex models tend to approximate data better than simple ones. The same observation seems warranted for any pair of conflicting criteria for model assessment.

Optimal model selection has been studied primarily within the automated machine learning community [3, 4, 5], with a common point of departure being that model selection can be framed as a sequential decision making problem. Decisions include the choice of algorithms (e.g., linear regression), methods (e.g., imputation of missing data) and hyperparameters [6]. The goal is to find sequences, and therefore models, that minimize a loss function. Hutter et al. [3] point out the difficulty in applying classical optimization to such a problem, citing as their main reasons (a) the complexity of the search space and (b) the fact that little is known about the loss's analytical properties, say convexity, when viewed as a function of the decisions. Accordingly, Feurer et al. [7] explore the search space using Bayesian optimization (BO) [8], which involves repetitive sampling of sequences around the current best sequence, based on performance estimates provided by a surrogate. Rokatoarison et al. [9] use Monte Carlo tree search [10] for structures, i.e algorithms and methods, and BO for hyperparameters. Structures are built incrementally by "walking down" a tree, adding one algorithm or method at a time, whereas hyperparameters are sampled as previously. Other works have focused on particular areas of model selection. For instance, Khurana et al. [11, 12] propose automatic feature engineering algorithms for regression and classification. Their goal is to determine, based on a predefined set of functions, what sequence of variable transformations yields the most predictively accurate model when applied on a dataset. For this they define a "transformation tree" in which nodes represent transformed datasets and edges represent functions, then they compare *ad hoc* search heuristics [11] with strategies obtained through reinforcement learning [12].

Zoph et al. [13] also leverage reinforcement learning, but do so in the context of tuning a convolutional neural network's hyperparameters. Their method involves training a recurrent neural network to select hyperparameters in such a way as to maximize expected model accuracy.

A drawback to the general approach represented by the aforementioned works is that it is single-objective, whereas model selection is essentially multi-objective. Here we propose a different approach, in which more than one metric is considered. We assume that the metrics conflict; that is, there exists no solution that optimizes all of them simultaneously, in which case the expert would be prepared to make tradeoffs. Thus we are primarily interested in identifying the set of models that cannot be improved in any metric without causing the others to deteriorate; this is an adaptation of a core Pareto (or multi-objective) optimization concept known as the Pareto front [14, 15, 16].

Albeit equivalent, models which lie on the Pareto front can be seen as representing different tradeoffs between the objectives. From a decision making perspective, therefore, the principal appeal of our approach is that it focuses the expert's attention on a set of equally good alternatives that they can ultimately choose from based on their preferences and the value placed on each objective [17, 18]. Another appeal is that by examining the shape of the Pareto front, the expert is able to gauge how much compromise is needed from some metric to improve others [14], and is hence able to locate regions in the model space where small concessions in some metrics permit considerable improvements in others. From a statistical perspective, our approach may assist in improving our understanding of the relationships among metrics and in informing theoretical investigations in that direction.

In this article, we present a model selection algorithm which searches for Pareto optimal models of a continuous response variable, given a dataset and a set of performance metrics. It does this by efficiently exploring a *model search graph*. We devise and implement a search heuristic which draws on ant colony optimization (ACO), a combinatorial optimization technique modeled after ant colonies [19]. ACO's first application was the traveling salesperson problem [20], but its scope has expanded tremendously in the last two decades, with applications in experiment design [21], project scheduling [22], spatial clustering [23],

hydraulic engineering [24] and healthcare [25]. Moreover, various extensions of it have been developed for Pareto optimization [26, 27], one of which was used in our work.

The remainder of the article is organized as follows. Section 2 states the combinatorial optimization problem underpinning the article, provides some terminology and introduces the model search graph. Section 3 describes a generic ACO algorithm and shows how it can be adapted to Pareto optimization. Section 4 presents our algorithm. On the basis of experiments, Section 5 compares the algorithm with AIC-based stepwise regression [28, 29], and sheds light on some of its basic properties.

# 2 Model Search Graph

We concentrate on linear regression models of a response $Y$ that may feature powers of variables $X_1, ..., X_p$, $p \geq 1$, together with two-way interactions. For $p = 3$ and a maximum exponent $e_{max} = 2$, these models have as an upper bound

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1^2 + \beta_5 X_2^2 + \beta_6 X_3^2 + \beta_7 X_1 X_2 + \beta_8 X_1 X_3 + \beta_9 X_2 X_3 + \epsilon \quad (1)$$

This work is an attempt to construct an efficient procedure for building satisfactory models of this form, in a way similar to forward selection but one which is stochastic and multi-objective. We first show that building such models is equivalent to traversing a graph.

## 2.1 Models as Paths on a Graph

Given $p$ and $e_{max}$, $M(p, e_{max})$ will denote the set of models described above. This set is finite. Let again $p = 3$ and $e_{max} = 2$. There are 8 ways of choosing two-way interactions: (1) $X_1X_2$; (2) $X_1X_3$; (3) $X_2X_3$; (4) $X_1X_2$ and $X_1X_3$; (5) $X_1X_2$ and $X_2X_3$; (6) $X_1X_3$ and $X_2X_3$; (7) $X_1X_2$, $X_2X_3$ and $X_1X_3$; or (8) no interactions. There are 8 ways of selecting the first-power terms, and 8 ways of selecting the second-power terms, including the possibility of not selecting any. This gives a total of 512 models. In general, $|M(p, e_{max})| = 2^{N_p + p \cdot e_{max}}$, where $N_p = \binom{p}{2}$.

The directed graph in Figure 1 contains $M(3, 2)$, with each model corresponding to exactly one full path. We will call it a model search graph. It can be generalized for any $p$

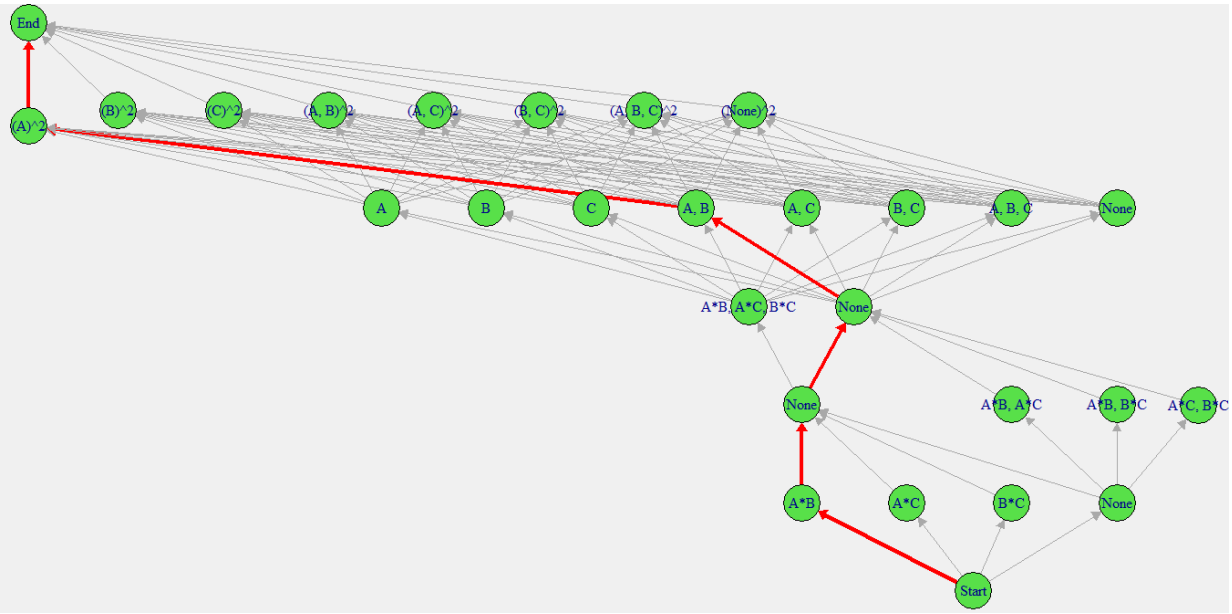


Figure 1: Model search graph for $p = 3$ and $e_{max} = 2$, with A $= X_1$, B $= X_2$ and C $= X_3$. The red path corresponds to the model $Y = \beta_0 + \beta_1 A + \beta_2 B + +\beta_3 A^2 + \beta_4 AB + \epsilon$.

and $e_{max}$ by following the same construction principles as above; that is, by enumerating the possible combinations of interaction terms, of power-of-one terms, and so on. Some comments on the graph are supplied in the online Appendix C.

## 2.2 Problem Statement

Let $F_1$, ..., $F_Q$ be $Q$ performance metrics to be minimized with respect to a dataset $D$. For fixed $p$ and $e_{max}$, a combinatorial Pareto optimization problem follows:

$$\min_{x \in M(p, e_{max})} \quad (F_1(x, D), ..., F_Q(x, D)) \tag{2}$$

where $F_q(x, D)$, $q \in \{1, \ldots, Q\}$, denotes the value of $F_q$ for a model $x \in M(p, e_{max})$ with respect to $D$.

In terms of the model search graph, and assuming as in the Introduction that the metrics conflict, solving (2) amounts to finding the full paths corresponding to models that are Pareto optimal with respect to $F_1, \ldots, F_Q$ (Definition 1 and 2). The Pareto front [18] is the set of such models.

**Definition 1** *Let $x$ and $y$ be models in $M(p, e_{max})$. $x$ is said to dominate $y$ on $D$ if $F_q(x, D) \leq F_q(y, D)$ for all $q \in \{1, \ldots, Q\}$ and $F_k(x, D) < F_k(y, D)$ for some $k \in \{1, \ldots, Q\}$.*

**Definition 2** *Let $x$ in $M(p, e_{max})$. $x$ is Pareto optimal (or nondominated, or efficient) if, for all $y \in M(p, e_{max})$, $y$ does not dominate $x$.*

We would like to point out, in passing, that the definitions above make no reference to the character of the relationships among the metrics. Indeed, even with nonconflicting metrics, one can still talk of a Pareto front, namely the singleton containing the model that minimizes all the metrics at once. In this case, however, the front's interest is clearly reduced, and Problem (2) could in principle be solved using single-objective optimization techniques.

Computing the Pareto front may be intractable [17]. Only an exact algorithm could determine with certainty whether a model is Pareto optimal, but such an algorithm typically proceeds by implicit enumeration of the search space [30], which makes it unsuitable for large graphs. This motivates the interest in heuristics, which are computationally efficient techniques for generating good solutions [31]; that is, solutions that are "*feasible and significantly better than a solution that would have been designed by a human expert*" [30]. Constructive heuristics, meaning those that build solutions stepwise, are known to be the fastest [19, 32], but they also seem to be the most relevant to the objective we outlined in introducing this section. Even more relevant are algorithms based on ant colony optimization [19], which are designed specifically for problems that can be reduced to finding desirable paths in a graph.

It is important to note that most applications of Pareto optimization aim not at computing the Pareto front but rather at approximating it [33]. Typically, a good Pareto front approximation will be a set of solutions in which (a) no solution dominates another; (b) the solutions are close to those on the true front; and (c) the solutions are diverse [15, 34, 35]. Perhaps the best known measure of the quality of a Pareto front approximation that accounts for (b) and (c) is the hypervolume, which is the size of the objective space enclosed by the points on the front and a dominated reference point [36]. The hypervolume has

the property of being strictly Pareto compliant [34]; that is, if $A$ and $B$ are Pareto front approximations such that the points in $B$ are dominated by those in $A$, then $A$ has a higher hypervolume than $B$. This observation will be useful in Section 5, where we compare different algorithm settings based on the Pareto front approximations obtained in each one.

# 3 Multi-objective Ant Colony Optimization

We now turn to ant colony optimization, an umbrella term for a group of algorithms in which artificial agents imitate the behavior of foraging ants to solve combinatorial optimization problems [19]. The basis for ACO is the assumption that simulation of stigmergy, i.e the discharging and following of pheromones by insects, can solve problems analogous to finding the shortest path in a graph. ACO deals traditionally with problems involving a function to be minimized with respect to a set of discrete decision variables subject to certain feasibility constraints. We will see, however, that the underlying ideas can easily be extended to the case where multiple functions are to be minimized.

ACO ants construct solutions incrementally and independently by traversing a graph whose edges represent value assignments for decision variables. An edge $(i, j)$ is described at time $t$ by the amount of pheromones present on it, $\tau_{ij}(t)$, and a so-called "heuristic value" (HV), $\eta_{ij}$ [19]. Pheromones are "discharged" during travel, with paths closer to optimal solutions receiving greater amounts of pheromones. Hence, $\tau_{ij}(t)$ is a cumulative indicator of the quality of solutions which had been built up to time $t$ using $(i, j)$. $\eta_{ij}$ expresses an *a priori* belief about whether $(i, j)$ belongs to an optimal solution. When approaching the TSP, for example, it is customary to let $\eta_{ij} = \frac{1}{d_{ij}}$, $d_{ij}$ being the distance between cities-nodes $i$ and $j$ [37]; the intuition is that a short tour can be obtained by repeatedly going to the closest yet-to-be-visited city until no city is left.

An ant chooses its destinations as a function of the pheromones and HVs of the edges available. Regions with relatively high concentrations of pheromones tend to be preferred [21]. Once a complete solution has been built, it is evaluated and pheromones are deposited on its constituent edges. The amount of pheromones deposited is commensurate with the

quality of the solution. Additionally, pheromones "evaporate" with time. Ideally this prevents a too rapid convergence to suboptimal regions, and encourages exploration [19, 38].

A multi-objective version of ACO was proposed by [26] in which $\tau_{ij}(t)$ and $\eta_{ij}$ are $Q$-sized vectors, where $Q$ is the number of objectives, such that $\tau_{\mathbf{ij}}(t) = (\tau_{ij,1}(t), ..., \tau_{ij,Q}(t))$ and $\eta_{\mathbf{ij}} = (\eta_{ij,1}, ..., \eta_{ij,Q})$. Ants evaluate their solutions and deposit a separate amount of pheromones for each objective. For all $(i, j)$ and $q \in \{1, \ldots, Q\}$, the $q$-th component of $\tau_{\mathbf{ij}}(t)$ is updated as

$$\tau_{ij,q}(t+1) = (1-\rho) \cdot \tau_{ij,q}(t) + \sum_{k=1}^{m} \Delta\tau_{ij,q}^{k} \tag{3}$$

where $\rho \in [0, 1]$ is the evaporation rate; $m$ the number of ants in the colony; $\Delta\tau_{ij,q}^{k}$ the amount of pheromones deposited by ant $k$ with respect to $F_q$, which measures solution quality with respect to $F_q$. The $Q$ updates occur simultaneously.

For concreteness, suppose $Q = 2$. Assuming ant $k$ occupies node $i$ in the construction graph, it will move to a successor node $j$ with a probability

$$p_{ij}^{k} = \begin{cases} \frac{(\tau_{ij,1}^{\lambda_1} \cdot \tau_{ij,2}^{\lambda_2})^{\alpha} (\eta_{ij,1}^{\lambda_1} \cdot \eta_{ij,2}^{\lambda_2})^{\beta}}{\sum_{p \in S(i)} (\tau_{ip,1}^{\lambda_1} \cdot \tau_{ip,2}^{\lambda_2})^{\alpha} (\eta_{ip,1}^{\lambda_1} \cdot \eta_{ip,2}^{\lambda_2})^{\beta}} & (i,j) \in N(s^k) \\ 0 & (i,j) \notin N(s^k) \end{cases} \tag{4}$$

where $s^k$ denotes the path that $k$ has traveled so far; $N(s^k)$ the set of edges that can be added to $s^k$ without violating the problem's constraints; $S(i)$ the successors of $i$; $(\alpha, \beta)$ parameters controlling the importance of an edge's pheromones relative to its HVs; and where $\lambda_q \in [0, 1]$ is the weight of $F_q$ relative to the other objective, such that $\lambda_1 + \lambda_2 = 1$. Time was dropped from the expression for brevity.

# 4 Proposed Algorithm

We now describe an algorithm for solving Problem (2). We will call it MOMSACO, which is shorthand for *Multi-objective Model Selection based on Ant Colony Optimization.* We treat the model search graph as a construction graph for a colony of size $m$. For simplicity, a Pareto front approximation will henceforth be referred to as a Pareto front. To avoid ambiguity, the Pareto front will be referred to as the true Pareto front. We use the notation of Section 2 and 3.

## 4.1 Initialization

The algorithm requires a dataset in which the columns representing $Y$ and $X_1, ..., X_p$ were designated. The dataset is randomly partitioned into a training dataset, $D_{train}$, and a test dataset, $D_{test}$. A value for $e_{max}$ is required for generating the model search graph.

HVs are initialized after the graph has been generated. Informally, $\eta_{ij}^{F_q}$ should be high if selection of $j$ is likely to bring an immediate improvement in $F_q$. In the TSP, for example, the biggest immediate improvement in traveled distance comes from visiting the closest city. It is hard to draw an analogy here, because model assessment metrics are not necessarily separable in the way distance is. Instead, if the current node is $i$, we infer the likely improvement in $F_q$ resulting from the selection of successor node $j$ based on a sample of models containing both $i$ and $j$. Specifically, we let, for $q \in \{1, \ldots, Q\}$,

$$\eta_{ij}^{F_q} = \frac{\Phi_i^{F_q}}{\overline{F_q}(., D_{test})_{ij}} \tag{5}$$

where:

- $\overline{F_q}(., D_{test})_{ij}$ is the mean of $F_q$, with respect to $D_{test}$, in a random sample of $n\_init$ models containing $(i, j)$. $n\_init \geq 1$ is a user-defined parameter.

- $\Phi_i^{F_q}$ is the maximum of $\overline{F_q}(., D_{test})_{il}$ over $l \in S(i)$.

## 4.2 Model Search

The $m$ ants construct models independently by walking from *Start* to *End*, selecting at each stage a node from the distribution defined by Equation (6). Let us describe this process in detail for some ant $k$, $k \in \{1, \ldots, m\}$. The ant is located initially in *Start*. A successor node is chosen after calculating $p_{ij}^k$, where $i = Start$, for all successors $j$. The ant joins this successor then continues in the same manner until it reaches *End*. Note that if an interaction node other than *None* was selected, the ant is required to choose *None* for its next moves until the next node is a power-of-one node (for an explanation, see the online Appendix C). Upon reaching *End*, a model will have been fully constructed; call it $m_k$. Its coefficients are estimated on $D_{train}$ then its metrics are evaluated on $D_{test}$. The

ant subsequently returns to *Start* by taking the opposite path. Along the way, it deposits a separate amount of pheromones for each metric, $\Delta\tau^k_{ij,q}$ ($q \in \{1, \ldots, Q\}$), on every visited edge. These are identical for all the edges taken. A percentage $\rho$ of the old pheromones evaporate. These steps occur at the same time colony-wide, and constitute a full iteration.

Conceptually, $\Delta\tau^k_{ij,q}$ should be a nonincreasing function of $F_q$. Hence, we take $\Delta\tau^k_{ij,q}$ to be the inverse of the value obtained for $F_q$ by ant $k$, i.e

$$\Delta\tau^k_{ij,q} = \frac{\delta_{ij,k}}{F_q(m_k, D_{test})} \tag{6}$$

where $\delta_{ij,k} = 1$ if $(i, j)$ was visited by $k$, and 0 otherwise. At the end of an iteration, $\tau_{ij,q}$ is updated for all $q \in \{1, \ldots, Q\}$ according to

$$\tau_{ij,q}(t+1) = (1-\rho) \cdot \tau_{ij,q}(t) + \sum_{k=1}^{m} \frac{\delta_{ij,k}}{F_q(m_k, D_{test})} \tag{7}$$

### 4.3 Populating a Pareto Front

A Pareto front is formed as iterations pass. To form the initial front, we compare the first iteration's models, then we drop the dominated ones. The remaining models could be Pareto optimal, so they constitute the initial front. From the second iteration onward, new models are compared with each other then with the present front. If a new model is not dominated by any of the others with which it was compared, then it is added to the front. However, if a model on the front turns out to be dominated by some new model, then it is removed. This procedure ensures that out of all the models found up to the present, only those that have not been proven to be dominated compose the front.

## 5 Results and Analysis

We begin this section with an experimental comparison of AIC-based stepwise regression (henceforth AIC-SR) [28] and MOMSACO, the two being sequential, multi-objective procedures for model selection (Section 5.1). Section 5.2 through 5.4 deal with some salient properties of MOMSACO.

In all the experiments reported, the metrics to be minimized were the root-mean-square error (RMSE) and the number of coefficients (NoCoeff). RMSE indicates accuracy because it captures the differences between a model's predictions and the values observed for the response [39]. NoCoeff describes a model's simplicity. The relevance of a Pareto front-based approach stems from the observation that it is usually impossible to minimize RMSE and NoCoeff simultaneously: when NoCoeff is minimal, i.e when there are no variables in the model, RMSE is generally suboptimal. One exception is when the response is constant, in which case the intercept-only model would score an RMSE of zero.

To initialize MOMSACO, eight models per edge were drawn uniformly ($n\_init = 8$), and Equation (7) was used. Initial pheromones were of one per metric per edge. Experiments spanned 100 iterations, and were run 50 times due to the randomness of navigation. For each dataset, 70% of the observations were used for training, and 30% for calculating the metrics. There were 15 ants. In Section 5.1, $\rho$ was set at 0.5, $\lambda_1$ and $\lambda_2$ at 0.5, $\alpha$ and $\beta$ at 1. A Pareto front's hypervolume was the total area of the rectangles enclosed by the points on the front and a reference point. This point's location varied with each dataset. Details on how the reference points and hypervolumes were determined are supplied in the online Appendix A. In Section 5.4 and 5.5, the term "transition probability" refers to the quantity defined by Equation (6). The experiments were run in R, on a laptop with an Intel i3 processor running at 2.10GHz and using 4GB of RAM.

## 5.1 Examples Where AIC-SR Fails But MOMSACO Does Not

The motivation for comparing MOMSACO with AIC-SR is threefold: (1) they are both multi-objective procedures; (2) they differ in fundamental ways; and (3) if cases exist in which MOMSACO outperforms AIC-SR, then positive conclusions can be drawn about its usefulness. Perhaps the most conspicuous difference between the two lies in the nature of the procedure. AIC-SR is deterministic, subject to a fixed set of rules based on statistical tests; MOMSACO is stochastic, iterative and one that "learns" over time. Secondly, consider the way in which candidate models are compared. AIC-SR compares models on the basis of AIC, with lower values preferred [29]. In contrast, MOMSACO does not score models;

rather, it compares their metrics then discards those that are dominated. The former mode of comparison produces a ranking of models; the latter supplies a range of models, each of which represents a particular compromise among the metrics being studied. The latter also allows some awareness of the relationships among the metrics; in particular, it helps gauge the extent to which they conflict, and identify opportunities for substantial improvements in one metric that require small concessions in another. Awareness of such relationships, which is important to decision making, is precluded by the first mode of comparison.

We now give examples of synthetic datasets where (a) MOMSACO outperforms AIC-SR in the sense of being able to find data-generating models, and (b) comparing models according to Pareto dominance is preferable to ranking them according to AIC. Such datasets abound, but for the sake of exposition we focus on two. Both were comprised of variables $X_1$, $X_2$, $X_3$ and $X_4$ and a response $Y$. In $D_1$, the four variables were drawn independently from normal distributions; in $D_2$, they were drawn from a multivariate normal distribution with $X_1$ and $X_2$ correlated. $Y$ was generated in each case according to

$$Y = 2 + X_1 + 1.5X_2 + X_1X_2 + X_4^2 + \epsilon \tag{8}$$

where $\epsilon$ was drawn from the standard normal distribution. We will henceforth refer to (10) as the true model. $X_1$, $X_2$, $X_3$ and $X_4$ were all allowed to feature in the models, so $p$ was equal to four. $e_{max}$ was taken to be two. This produced a search graph with 16 384 models. Details on the simulation of $D_1$ and $D_2$ (and on a third, unreported example) are provided in the online Appendix B.

To substantiate (a) and (b), we generated $N = 1000$ observations in $D_1$ and 10 000 in $D_2$, then we ran two experiments per dataset. In the first, we computed the AIC value of every model in the graph, then we ran a full search of $M(4, 2)$ to find the true Pareto front with respect to RMSE and NoCoeff. In the second, we ran forward, backward and bidirectional AIC-SR, then we initialized MOMSACO and ran 50 trials of it. An aggregate Pareto front was then obtained by combining the fronts of the MOMSACO trials and discarding the dominated models. Figure 3 shows both aggregate fronts. To ensure a fair comparison, we required AIC-SR's scope to be the same as MOMSACO's, i.e $\{X_1, X_2, X_3, X_4, X_1^2, \ldots, X_4^2, X_1X_2, \ldots, X_3X_4\}$.

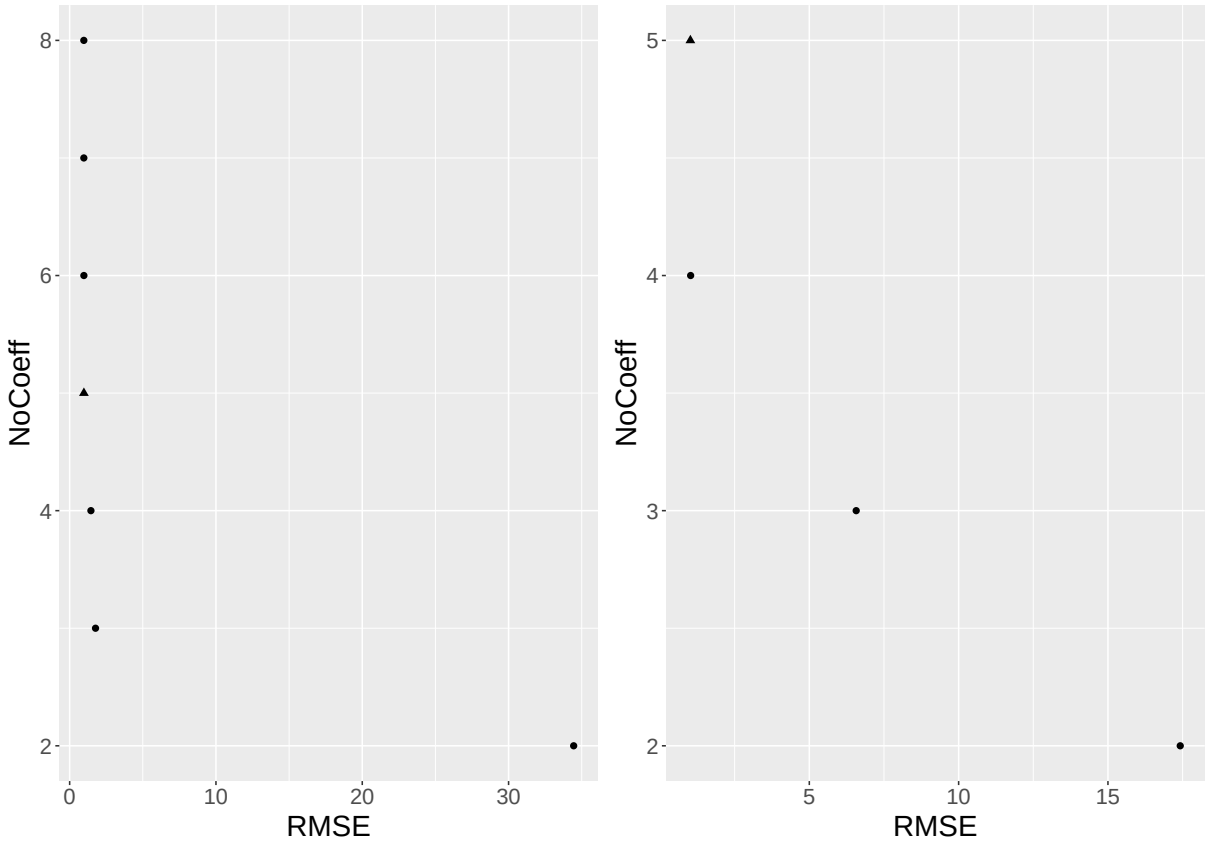


Figure 2: The aggregate Pareto fronts obtained for $D_1$ (N = 1000; left) and $D_2$ (N = 10 000; right). Triangles represent the true models.

Table 1: Effect of sample size and correlation on the ability of both algorithms to find the true models. Y denotes success in finding these models, and N denotes failure.

| **Sample size** | 1000 | 10 000 | 100 000 |
|---|---|---|---|
| $D_1$ (MOMSACO) | Y | Y | Y |
| $D_1$ (AIC-SR) | N | Y | Y |
| $D_2$ (MOMSACO) | Y | Y | Y |
| $D_2$ (AIC-SR) | N | N | N |

In both datasets, the true model was truly Pareto optimal with respect to RMSE and NoCoeff, but suboptimal with respect to AIC. Several models had better AIC values than the true one. This suggests that Pareto dominance is a sound basis for comparing models when RMSE and NoCoeff are its metrics. It also shows that there are situations in which a ranking of models based on AIC can be misleading in that it places data-generating models below others. We also found that whereas AIC-SR did not succeed in finding the true model, MOMSACO did. This was true for both datasets (see the triangles in Figure 3), which proves that MOMSACO can outperform AIC-SR.

Illustrations of how Pareto dominance can inform model selection are provided by both fronts' bottom two models (Figure 3), which show that a small concession in NoCoeff (from two variables to three) allows for significant gains in RMSE (from approximately 34.4 to 1.8 in $D_1$). Both Pareto fronts also give clear insight into the tradeoff represented by each model, an insight not provided by AIC values.

Additional experiments with $D_1$ and $D_2$ showed that the relative performance of AIC-SR was affected by the sample size and the presence of correlated predictors (Table 1). For $N = 10\,000$, AIC-SR was finally able to retrieve the true model in $D_1$. This was not observed in $D_2$, which differs with $D_1$ only in that it was generated by a model with correlated predictors. This suggests the following: (a) in the absence of correlated predictors, AIC-SR requires a larger sample than MOMSACO to find the data-generating model; (b) when correlations exist, AIC-SR may require a much larger sample, which however was not confirmed by the experiments, because AIC-SR failed even with $N = 10^6$. In any event,

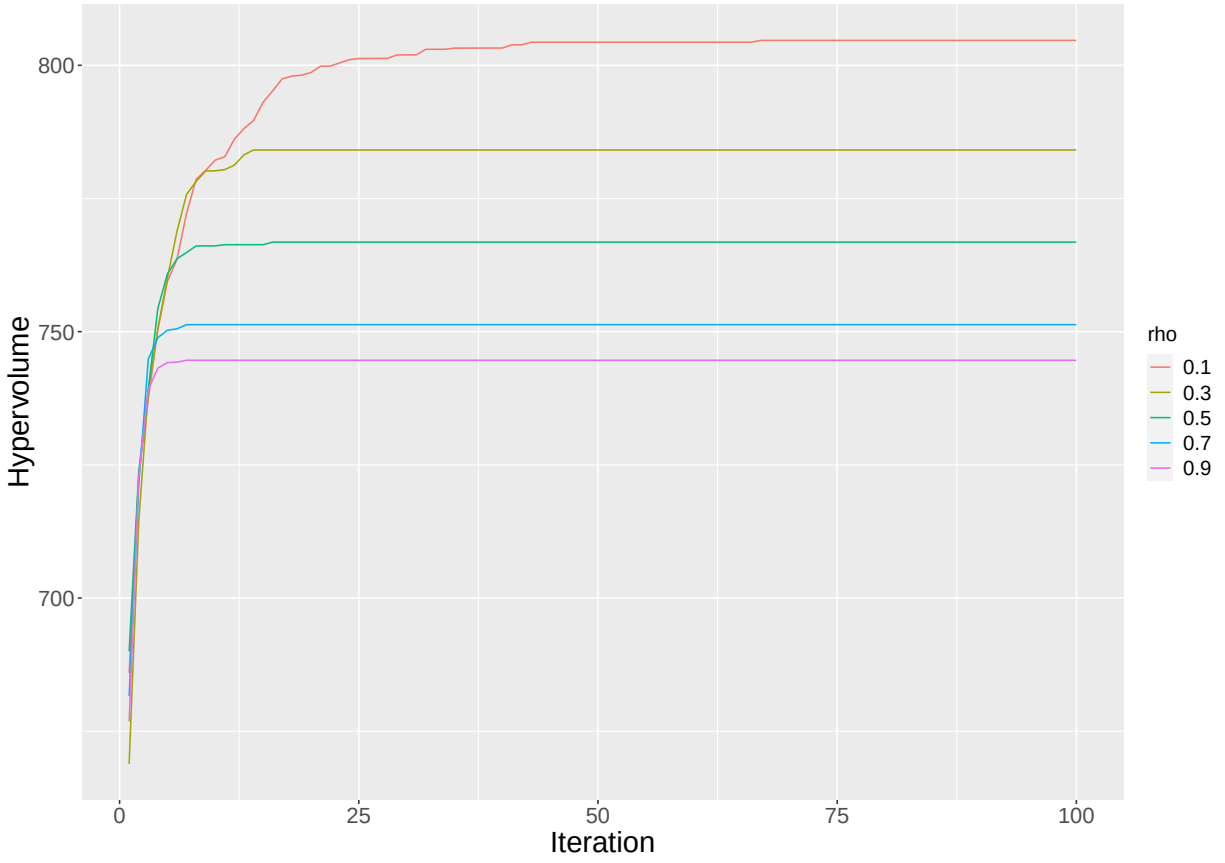


Figure 3: Evolution of the hypervolume for various evaporation rates (averaged in each case over the 50 trials). Dataset: $D_1$ ($N = 1000$). Should be viewed in color.

it seems that in the presence of correlations, MOMSACO can recover the data-generating model regardless of the sample size.

## 5.2 Pareto Fronts

Now we concentrate solely on MOMSACO. In each of the previous experiments, the Pareto front's size and hypervolume kept increasing until they leveled out. Figure 4 depicts this process for $D_1$ ($N = 1000$) for various evaporation rates, omitting the fronts' sizes for clarity. In general, the hypervolume improved most rapidly within the first few iterations. This was when the fronts were being populated at the fastest rate (Table 2).

While work is needed to pinpoint the effect of each parameter on the Pareto front, experiments suggested that evaporation is an important factor. Notably, the hypervolume

Table 2: Pareto front size per iteration (averaged over the 50 trials). Dataset: $D_1$ ($N = 1000$).

| **Iteration** | **Avg. no. of models on the Pareto front** |
|---|---|
| 1 | 3.6 |
| 3 | 4.6 |
| 12 | 5.8 |
| 15 | 6.0 |
| 24 | 6.3 |
| 100 | 6.3 |

plateaued earlier for larger evaporation rates, but attained higher values for smaller ones. This means that for smaller values of $\rho$, the discovery of new nondominated models spanned a longer period of time, and higher quality fronts were formed.

It is noteworthy that, on average, MOMSACO was able to recover 26.54% of the true Pareto front (see the rates of contribution in Table 3) by evaluating as few as 219 models, i.e 1.33% of the total size of the search graph.

## 5.3 Ant Navigation

Figure 4 describes the evolution of Pareto front quality over time, using the hypervolume as an indicator. It may be interesting, from an optimization point of view, to ask how the mean of each metric evolves while the front is being populated. Figure 5 shows that as the search was proceeding, there was a gradual decline, and therefore improvement, in the average values of RMSE and NoCoeff among models on the Pareto front. This indicates that both metrics were being optimized.

The improvement observed in Figure 4 and 5 can be attributed to the movement of the pheromones. Initially, these do not contribute to navigation because they are equally distributed among the edges. Rather, it is the HVs, which differ from an edge to another, that determine the initial transition probabilities. Therefore, the ants will most likely start by taking the paths with the highest HVs. Later, the pheromones begin to vary as a result

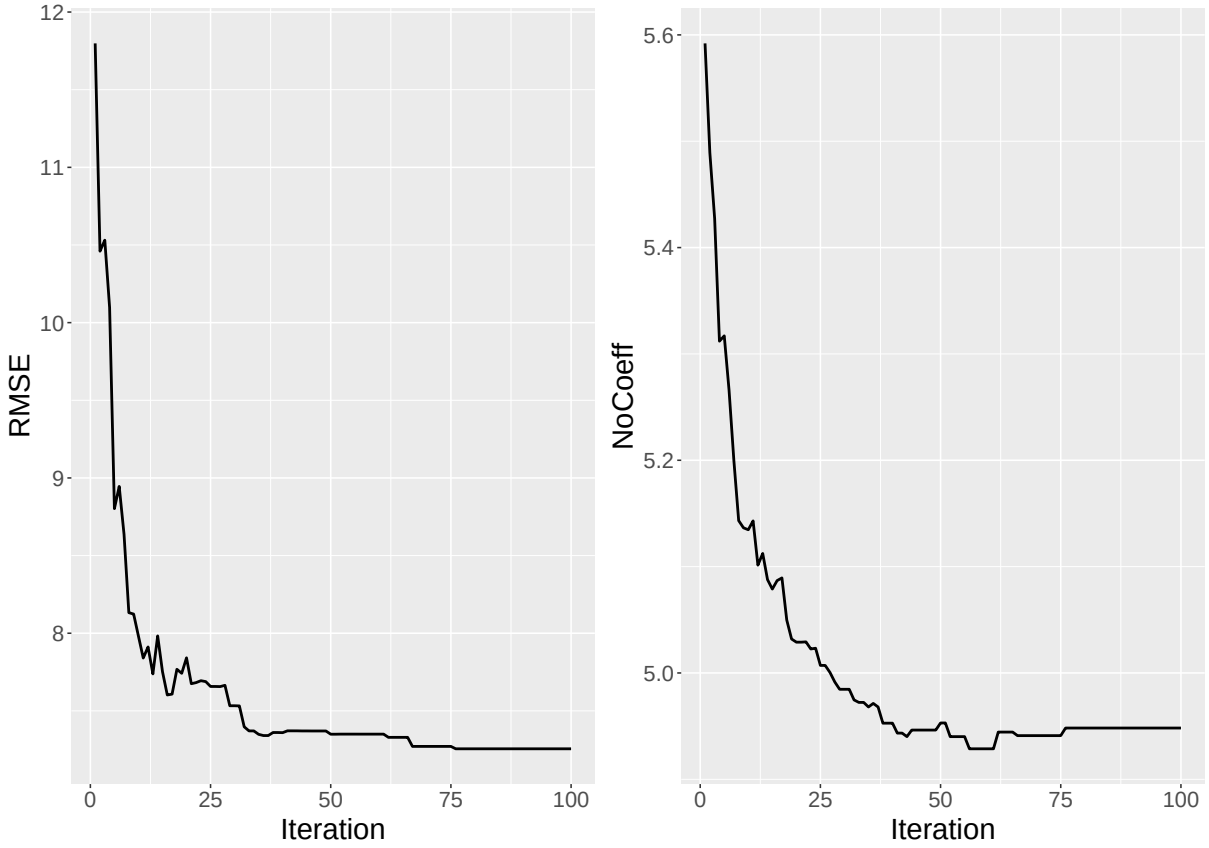


Figure 4: Evolution of the Pareto front's average RMSE and NoCoeff values (averaged over 50 trials). Dataset: $D_1$ ($N = 1000$).

Table 3: Pareto front data after 100 iterations for different evaporation rates. Figures indicated are averages (standard deviations) over the 50 trials. The rate of contribution was defined as the percentage of models on the true Pareto front that belonged to the MOMSACO front for which the rate was computed. Dataset: $D_1$ ($N = 1000$).

| **Evaporation rate** | **0.1** | **0.3** | **0.5** | **0.7** | **0.9** |
|---|---|---|---|---|---|
| Hypervolume | 804.68 (18.44) | 784.11 (23.56) | 766.81 (27.75) | 751.32 (28.68) | 744.63 (28.87) |
| Pareto front size | 6.30 (0.76) | 5.76 (1.06) | 5.48 (1.01) | 5.04 (1.03) | 4.14 (1.14) |
| Proportion of Pareto optimal models | 46.85% (16.25%) | 29.61% (18.65%) | 16.27% (19.45%) | 11.93% (17.26%) | 16.20% (21.30%) |
| Rate of contribution to the true Pareto front | 26.54% (8.58%) | 15.09% (9.65%) | 8.18% (10.28%) | 5.09% (7.15%) | 4.91% (5.57%) |

of their deposition and evaporation. This affects navigation by altering the probabilities of the traversed edges, so that those leading to lower metrics become more probable.

## 5.4 On Initialization

Since the HVs determine the initial transition probabilities, they significantly alter the behavior of the colony in the first iteration. MOMSACO's performance in subsequent iterations is also, by that very fact, dependent on the HVs. The method by which these are computed is a critical component of ACO algorithms in general, and of MOMSACO in particular.

To evaluate the initialization scheme described in Section 4.1, further experiments were run (1) with an arbitrary assignment of HVs (50 trials), then (2) with no HVs at all, i.e $\beta = 0$ in Equation (6) (50 trials). Figure 6 juxtaposes the mean hypervolumes of both sets of experiments with that of the original $D_1$ experiment ($N = 1000$). We can see that in the original experiment, i.e when the HVs had been set using Equation (7), the Pareto front was of a consistently superior quality, and was thus closer to the true front. When the HVs were initialized arbitrarily, the resulting Pareto front was the poorest. Two conclusions

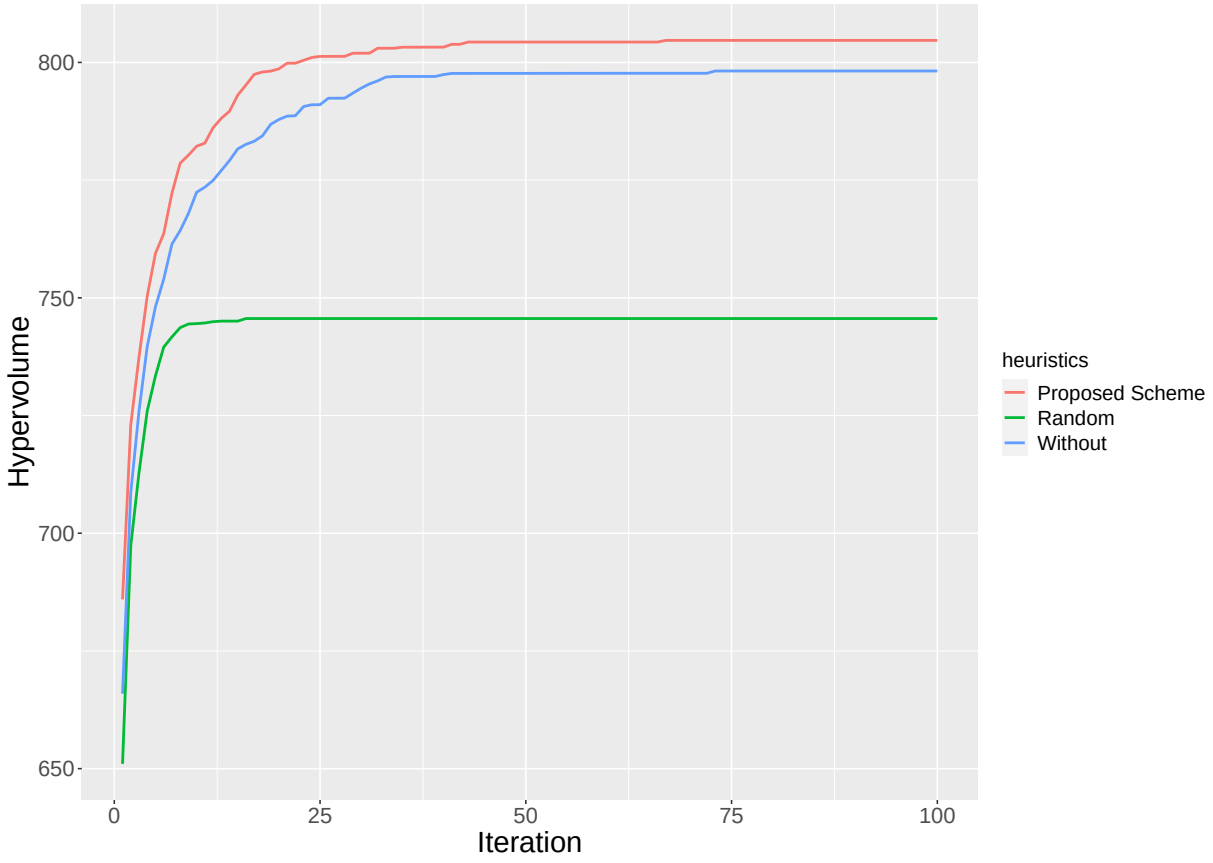


Figure 5: Evolution of the hypervolume when (a) the HVs were initialized under the scheme of Section 4.1, when (b) they were initialized randomly, and when (c) they were not present. Hypervolume was averaged over the 50 trials. Dataset: $D_1$ ($N = 1000$). Should be viewed in color.

arise from this: (a) our initialization procedure can enhance model search in that it allows the formation of higher quality fronts; and (b) HVs can hamper performance if they are not chosen carefully.

# 6 Conclusion

A model selection algorithm was developed that searches automatically for Pareto optimal linear regression models, given a dataset and a set of performance metrics. Candidate models combined two-way interactions with powers of the original explanatory variables.

The task of finding Pareto optimal models was framed as that of finding the corresponding paths on the model search graph. A search strategy was proposed that draws on ACO, a probabilistic technique well suited for problems involving graphs.

Experiments with simulated data showed that MOMSACO could recover the underlying models where AIC-SR could not. They also showed that these were actually suboptimal with respect to AIC, but Pareto optimal with respect to RMSE and NoCoeff. Only by increasing the sample size or reducing the correlations in the data-generating model was AIC-SR able to find it. These results suggest that MOMSACO is better suited for correlated predictors and small datasets than AIC-SR, and that comparing models according to Pareto dominance can inform model selection.

By juxtaposing the Pareto front's rate of contribution with the total number of models explored, we were able to conclude that MOMSACO can efficiently navigate the model search graph. We also found our initialization method to be a desirable component of MOMSACO because it enabled the formation of a higher quality Pareto front.

MOMSACO can be used as a variable selection procedure for potentially any type of regression model. Because it does not presuppose the nature of the models it is searching for, and because these are trained independently of it, the procedure can apply equally to non-linear models. If hyperparameters are required, however, then grid search or Bayesian optimization can supplement MOMSACO.

A number of questions remain. One is how the parameters should be set, given some reference point for hypervolumes, in order to obtain the best possible Pareto fronts. This is in itself an optimization problem, and a complex one, primarily because the data is a factor in the optimality of a configuration of parameters. We intend to study the matter in detail, hoping to enunciate some guidelines on how the parameters should be chosen in broad enough situations. A second question is how to better the representation of $M(p, e_{max})$ so that computational issues associated with large $p$ and $e_{max}$ could be alleviated. For large $p$, the model search graph may not fit into the memory of a computer. Immediate solutions are to consider only interactions and power-of-one terms, or to subject the variables to preliminary dimensionality reduction via techniques like principal component analysis [40].

A third question is how the current initialization procedure could be improved. We think incorporating some form of past knowledge may be beneficial. If, for instance, datasets similar to the one being used are available, and models that performed well on them are known, then information about these can be leveraged – for example, by awarding relatively high values to the constituent edges of similar models in the model search graph. A database could be set up that stores key statistical characteristics of a large number of datasets and corresponding models, which characteristics could be used to measure the similarity between a dataset of interest and those in the database, and between the graph models and the database models. Datasets and models could be extracted from academic papers, open machine learning competitions, experts and other relevant sources.

## SUPPLEMENTARY MATERIAL

Online Appendix A supplies an analytic expression for the hypervolume of a two-dimensional Pareto front and explains how reference points were chosen in Section 5. Appendix B gives details about data generation in Section 5. Appendix C offers comments about the model search graph. An R script and a package demonstrating the use of MOMSACO are also provided. Comments on the two files are in Appendix D.

## COMPETING INTERESTS

The authors declare that they have no conflict of interest.

# References


[1] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001.

[2] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, May 2017.

[3] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automatic Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[4] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[5] Shubhra Kanti Karmaker, Md Mahadi Hassan, Micah J Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.

[6] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 847–855, New York, NY, USA, 2013. Association for Computing Machinery.

[7] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[8] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[9] Herilalaina Rakotoarison, Marc Schoenauer, and Michèle Sebag. Automated machine learning with Monte-Carlo Tree Search. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3296–3303. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[10] Guillaume Maurice Jean-Bernard Chaslot Chaslot. *Monte-carlo tree search*, volume 24. Maastricht University, 2010.

[11] Udayan Khurana, Deepak Turaga, Horst Samulowitz, and Srinivasan Parthasrathy. Cognito: Automated feature engineering for supervised learning. In *2016 IEEE 16th*

*International Conference on Data Mining Workshops (ICDMW)*, pages 1304–1307, Dec 2016.

[12] Udayan Khurana, Horst Samulowitz, and Deepak Turaga. Feature engineering for predictive modeling using reinforcement learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 3407–3414, 2017.

[13] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. 2017.

[14] Yongtao Cao, Byran J. Smucker, and Timothy J. Robinson. A hybrid elitist Pareto-based coordinate exchange algorithm for constructing multi-criteria optimal experimental designs. *Statistics and Computing*, 27(2):423–437, mar 2017.

[15] Susan Wei and Marc Niethammer. The fairness-accuracy Pareto front. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(3):287–302, 2022.

[16] Felix J. Bierbrauer and Pierre C. Boyer. The Pareto-frontier in a simple Mirrleesian model of income taxation. *Annals of Economics and Statistics*, (113/114):185–206, 2014.

[17] Jessica L. Chapman, Lu Lu, and Christine M. Anderson-Cook. Impact of response variability on Pareto front optimization. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(5-6):314–328, 2015.

[18] Lu Lu, Christine M. Anderson-Cook, and Dennis K.J. Lin. Optimal designed experiments using a Pareto front search for focused preference of multiple objectives. *Computational Statistics  Data Analysis*, 71:1178–1192, 2014.

[19] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT Press, 06 2004.

[20] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernet-*

*ics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 26:29–41, 02 1996.

[21] M. Borrotti, G. Minervini, D. De Lucrezia, and I. Poli. Naïve bayes ant colony optimization for designing high dimensional experiments. *Applied Soft Computing*, 49:259–268, 2016.

[22] Lei Zhu, Jian Lin, and Zhou-Jing Wang. A discrete oppositional multi-verse optimization algorithm for multi-skill resource constrained project scheduling problem. *Applied Soft Computing*, 85:105805, 2019.

[23] Tülin İnkaya, Sinan Kayalıgil, and Nur Evin Özdemirel. Ant colony optimization based clustering methodology. *Applied Soft Computing*, 28:301–311, 2015.

[24] S.M. Bateni, M. Mortazavi-Naeini, B. Ataie-Ashtiani, D.S. Jeng, and R. Khanbilvardi. Evaluation of methods for estimating aquifer hydraulic parameters. *Applied Soft Computing*, 28:541–549, 2015.

[25] Kamil Krynicki, Javier Jaen, and Elena Navarro. An aco-based personalized learning technique in support of people with acquired brain injury. *Applied Soft Computing*, 47:316–331, 2016.

[26] Carlos García-Martínez, Oscar Cordón, and Francisco Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria traveling salesperson problem. *European journal of operational research*, 180(1):116–148, 2007.

[27] Daniel Angus and Clinton Woodward. Multiple objective ant colony optimisation. *Swarm intelligence*, 3(1):69–85, 2009.

[28] D. G. Kabe. Stepwise multivariate linear regression. *Journal of the American Statistical Association*, 58(303):770–773, 1963.

[29] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1998.

[30] Michel Bierlaire. *Optimization: Principles and Algorithms*. EPFL Press, Lausanne, 2nd edition, 2018.

[31] Ricardo García-Ródenas, José Carlos García-García, Jesús López-Fidalgo, José Ángel Martín-Baos, and Weng Kee Wong. A comparison of general-purpose optimization algorithms for finding optimal approximate experimental designs. *Computational Statistics  Data Analysis*, 144:106844, 2020.

[32] David Lai, Yijun Li, Emrah Demir, Nico Dellaert, and Tom Van Woensel. Self-adaptive randomized constructive heuristics for the multi-item capacitated lot sizing problem. *Computers & Operations Research*, 147:105928, 2022.

[33] F. Sambo, M. Borrotti, and K. Mylona. A coordinate-exchange two-phase local search algorithm for the d- and i-optimal design of split-plot experiments. *Computational Statistics and Data Analysis*, 71:1193–1207, 2014.

[34] Yongtao Cao, Byran J Smucker, and Timothy J Robinson. On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, 160:60–74, 2015.

[35] Peter Auer, Chao-Kai Chiang, Ronald Ortner, and Madalina Drugan. Pareto front identification from stochastic bandit feedback. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 939–947, Cadiz, Spain, 09–11 May 2016. PMLR.

[36] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer, 1998.

[37] Jürgen Branke and Michael Guntsch. Solving the probabilistic TSP with ant colony optimization. *Journal of Mathematical Modelling and Algorithms*, 3(4):403–425, Dec 2004.

[38] You Wan, Tao Pei, Chenghu Zhou, Yong Jiang, Chenxu Qu, and Youlin Qiao. Acomcd: A multiple cluster detection algorithm based on the spatial scan statistic and ant colony optimization. *Computational Statistics  Data Analysis*, 56(2):283–296, 2012.

[39] Robert Gilmore Pontius, Olufunmilayo Thontteh, and Hao Chen. Components of information for multiple resolution comparison between maps that share a real variable. *Environmental and Ecological Statistics*, 15(2):111–142, Jun 2008.

[40] Iain M. Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009. PMID: 20617121.