OR Spectrum (2024) 46:73–107
https://doi.org/10.1007/s00291-023-00716-0

ORIGINAL ARTICLE

One-stage product-line design heuristics: an empirical comparison

Daniel Baier[1] · Sascha Voekler[1,2]

Received: 6 April 2022 / Accepted: 23 March 2023 / Published online: 9 May 2023
© The Author(s) 2023

Abstract
Selecting or adjusting attribute-levels (e.g. components, equipments, flavors, ingredients, prices, tastes) for multiple new and/or status quo products is an important task for a focal firm in a dynamic market. Usually, the goal is to maximize expected overall buyers' welfare based on consumers' partworths or expected revenue, market share, and profit under given assumptions. However, in general, these so-called product-line design problems cannot be solved exactly in acceptable computing time. Therefore, heuristics have been proposed: Two-stage heuristics select promising candidates for single products and evaluate sets of them as product-lines. One-stage heuristics directly search for multiple attribute-level combinations. In this paper, Ant Colony Optimization, Genetic Algorithms, Particle Swarm Optimization, Simulated Annealing and, firstly, Cluster-based Genetic Algorithm and Max-Min Ant Systems are applied to 78 small- to large-size product-line design problem instances. In contrast to former comparisons, data is generated according to a large sample of commercial conjoint analysis applications (n = 2,089). The results are promising: The firstly applied heuristics outperform the established ones.

Keywords Product-line design · Conjoint analysis · Combinatorial optimization · Heuristics

✉ Daniel Baier
daniel.baier@uni-bayreuth.de

Sascha Voekler
sascha.voekler@b-tu.de

1 Chair of Marketing & Innovation, University of Bayreuth, Universitaetsstrasse 30, 95447 Bayreuth, Germany

2 Chair of Automation Technology, Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany

1 Introduction

Since its origins in the 1960s, conjoint amalysis (see,e.g. Debreu 1960; Luce and Tuckey 1964; Green and Rao 1969) has developed to a wide-spread marketing research tool for estimating customers' attribute-level partworths and choice models in dynamic markets. So, based on a survey among users of Sawtooth Software (probably the leader in conjoint analysis software, see sawtoothsoftware.com), each year more than 27,000 commercial conjoint analysis applications are performed world-wide, most of them in order to support firms with respect to product pricing, product redesign/repositioning, line extension, or new product introduction (see, e.g., Orme 2019; Baier and Kurz 2021). Roberts et al (2014) support these findings with their citation analysis as well as their surveys among researchers, mediators, and practitioners. They state that articles on conjoint analysis (including discrete choice analysis) had highest impact on marketing practice and that, from a practitioner's point of view, (new) product and brand management benefited the most from marketing research tools.

In order to further support such tasks based on results from a conjoint analysis application, over the years, a large number of product and product-line design problems were formulated to find "best" sets of attribute-level combinations. The problems vary with respect to the number of products to be designed (single vs. multiple new products in the line), the objective function (e.g., buyers' welfare or the focal firm's revenue, market share, and profit) and the assumed transformation of individual attribute-level partworths to choices (e.g., first-choice or logit). However, since the number of feasible solutions combinatorially depends on the number of new products R, the number of attributes K, and the number of levels per attribute L_k, many problem instances cannot be solved exactly in acceptable computing time. So, for nine new products, nine attributes, each with five levels, Complete Enumeration would require the evaluation and comparison of $\binom{5^9}{9}=\binom{1953125}{9}=1.14 \cdot 10^{51}$ feasible solutions. Kohli and Krishnamurti (1989) showed in their paper, that a typical problem, the shares-of-choices product-line design problem based on a first-choice assumption as choice rule is NP-hard. Many alternative algorithms have been proposed that vary with respect to the number of stages (one-stage vs. two-stage), the solution principle (exact vs. heuristic) and the solution method (e.g., Dynamic Programming Heuristic, Genetic Algorithms, Simulated Annealing).

Baier and Gaul (1999) provided an overview on these developments. From this overview, it becomes clear, that—till 1996—two-stage and one-stage heuristics were almost equally often proposed for product-line design. Two-stage heuristics (e.g. Green and Krieger 1985, 1987; Dobson and Kalish 1988, 1993) select in a first stage a rather small set of promising attribute-level combinations as candidates. Then, in a second stage, they evaluate candidate sets to receive "best" product-lines. The first stage reduces the overall solution space and can lead to suboptimal multiple attribute-level combinations, however—on the other side—it reduces the computing time. A typical approach for this first stage consists in selecting all attribute-level combinations that maximize at least one customer's utility (e.g. Green and Krieger 1987).

One-stage heuristics (see, e.g. Zufryden 1977; Kohli and Krishnamurti 1987, 1989; Kohli and Sukumar 1990) omit the reduction of the solution space by evaluating all feasible sets of attribute-level combinations as product-lines. Comparisons by Kohli and Sukumar (1990) as well as later by Belloni et al (2008a) empirically showed that one-stage heuristics outperform two-stage heuristics for this reason, especially when the first-choice assumption holds. Consequently, from 1995, one-stage heuristics have dominated the product-line design literature. In this paper, we, therefore, only discuss, apply, and compare one-stage heuristics and discuss related improvements.

However, it should be mentioned that recently great progress can be seen in assortment optimization (see, e.g., Rusmevichientong et al 2014; Sen et al 2018; Désir et al 2021), a research area that is closely related to two-stage product-line design: A retailer selects a number of shelf-space offers (a product-line) out of a large number of potential offers (candidates), so that her/his revenue is maximized under given shelf-space and/or cardinality constraints. The methodological advances in this research field nowadays allow to deal with rather large numbers of potential offers (up to 20, 200, 1000 according to Rusmevichientong et al 2014; Sen et al 2018; Désir et al 2021). However, still, the criticism of Kohli and Sukumar (1990) as well as Belloni et al (2008a) applies with respect to the reduction of the solution space to lines that consist of this still small number of candidates in contrast to the number of feasible candidates in a commercial product-line design problem. So, e.g., Selka (2013) shows in his summary of 2,089 commercial conjoint analysis applications, that 3 to 9 attributes with 2 to 7 levels are wide-spread, resulting in up to 7^9=40,353,607 feasible candidates. A reduction to a small number in a first stage (e.g. up to 1,000) is feasible, but according to Kohli and Sukumar (1990) as well as Belloni et al (2008a) may lead to suboptimal solutions.

This paper contributes to the product-line design research area by providing an overview on established problems and a comparison of recently developed one-stage heuristics to solve them. Moreover, three promising new heuristics (Cluster-based Genetic Algorithm, Max-Min Ant System, Max-Min Ant System with Local Search) are described and—for the first time—applied to product-line design problems. In contrast to former comparisons of product-line design heuristics, the problem instances in our comparison are generated according to characteristics of a large sample of commercial conjoint analysis applications (see Selka 2013; Selka and Baier 2014; Selka et al 2014).

The structure of this paper is as follows: In Sect. 2, the overview on product-line design problems is provided. Section 3 discusses one-stage heuristics to solve them, starting its overview from 1995 (as an update of Baier and Gaul 1999) and discussing results from former comparisons. Section 4 discusses selected promising one-stage heuristics based on these comparisons and introduces the new ones. For hyperparameter and parameter tuning, these heuristics are applied to 18 small- to large-size problem instances (with $5.1 \cdot 10^9$ to $1.6 \cdot 10^{153}$ feasible solutions). Finally, in Sect. 5, the tuned heuristics are applied to 60 additional problem instances (with $7.0 \cdot 10^5$ to $7.3 \cdot 10^{200}$ feasible solutions). The paper closes in Sect. 6 with conclusions and an outlook.

2 Product-line design problems: an overview

Up to now, marketing research and practice agree that the selection or adjustment of attribute-levels (e.g. components, equipments, flavors, ingredients, prices, tastes) for a firm's products based on conjoint analysis applications is an important task for marketers (see, e.g. Green and Krieger 1989, Kohli and Krishnamurti 1989, Nair et al. 1995, Balakrishnan and Jacob 1996, Chen and Hausman 2000, Steiner and Hruschka 2003, Schön 2010) Usually, the goal is to maximize expected overall buyers' welfare or firm's revenue, market share, or profit. Given are estimated partworths as well as assumed choice rules and contribution margins of consumers together with assumed attribute-levels of the competitors' current and/or future products. Formalized problems can be distinguished according to their objective function (e.g., buyers' welfare, revenue, market share, profit) and the choice rules assumed (e.g., deterministic and probabilistic choice). In the following, we shortly discuss the usual choice rule assumptions and—depending on their objective function—product-line design problems basing on them.

2.1 Choice rules

After performing a conjoint analysis application under the wide-spread linear-additive utility assumption (Wittink and Cattin 1989; Orme 2019), the marketing researcher receives estimated partworths β_{ikl} for a sample of I consumers (i=1,...,I) with respect to K attributes (k=1,...,K) and L_k levels (l=1,...,L_k). The partworths reflect the consumers' preferences for attribute-levels and can be used to evaluate J attribute-level combinations (products, candidates) from the consumers' point of view. So, with x_{jkl} as an indicator whether combination j (j=1,...,J) has level l of attribute k (=1) or not (=0), the utility u_{ij} of combination j for consumer i can be estimated as

$$u_{ij} = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \beta_{ikl} x_{jkl} \quad \forall i = 1,..., I;\ j = 1,..., J. \tag{1}$$

The straightforward assumption is that the combination with highest utility is always preferred and bought (deterministic choice rule). However, according to Green and Krieger (1988, 1992), due to data collection and estimation errors as well as diverted or less interested consumers, it can alternatively be assumed that other combinations are bought also with positive probabilities (probabilistic choice rule assumption).

2.1.1 Deterministic choice rule assumption

Especially in the product-line design literature, the first-choice rule is a wide spread choice rule assumption (see, e.g., Green and Krieger 1988, 1992). Consumer i is assumed to always select the combination that offers him the highest utility. The probability p_{ij} that consumer i selects combination j among the available J combinations is

$$p_{ij} = \begin{cases} 1, & \text{if } u_{ij} \geq u_{ij'} \quad \forall j' = 1,\ldots, J, \\ 0, & \text{else,} \end{cases} \qquad \forall i = 1,\ldots, I;\ j = 1,\ldots, J. \tag{2}$$

If two or more combinations have maximum utility (e.g., when having identical levels across all attributes), it is usually assumed that one of them is bought randomly. Alternatively, it can be assumed that they are bought with equally distributed probabilities (1 divided by the number of combinations with maximum utility). If we have stochastic partworths (e.g. as draws of a distribution, determined via Hierarchical Bayes estimation), the first-choice rule can also be applied to these draws. In this case, p_{ij} is estimated as the share of draws where j receives maximum utility (see, e.g., Baier and Polasek 2003; Hein et al 2022).

2.1.2 Probabilistic choice rule assumptions

The assumption that consumers always buy the combination with maximum utility has often been criticized in the conjoint analysis literature (see, e.g., Green and Krieger 1988; Louviere 1988). Even small differences among utilities lead to the prediction that the maximum utility combination is always bought, a contradiction to a complex data collection and estimation process with inaccurate and faulty partworths as results. Also, the focus on maximum utility could be wrong in a market where consumers buy combinations more or less randomly (e.g., when consumers have to choose among low involvement products or among more or less identical alternatives).

Probabilistic choice rules mitigate these problems by allocating probabilities to the combinations that (1) maintain the order of the utilities but (2) also allow to calibrate the choice rules according to market shares or past buying behavior. Two probabilistic choice rules are wide-spread in the conjoint analysis literature. The Bradley-Terry-Luce (BTL) choice rule (Bradley and Terry 1952; Luce 1959) assumes that probabilities p_{ij} are proportional to the utilities u_{ij}:

$$p_{ij} = \frac{u_{ij}^{\alpha}}{\sum_{j'=1}^{J} u_{ij'}^{\alpha}} \qquad \forall i = 1,\ldots, I;\ j = 1,\ldots, J(\alpha \geq 0). \tag{3}$$

The logit rule (McFadden 1976; Punj and Staelin 1978) assumes proportionality to exponentiated utilities:

$$p_{ij} = \frac{\exp(\alpha u_{ij})}{\sum_{j'=1}^{J} \exp(\alpha u_{ij'})} \qquad \forall i = 1,\ldots, I;\ j = 1,\ldots, J(\alpha \geq 0). \tag{4}$$

In both cases, the parameter α can be used to calibrate the choice rules. Values of α near or equal zero reflect an equal distribution of probabilities among available combinations, large values reflect the first-choice rule. Often, past purchase behavior of the consumers is used to estimate α as this is also done in pre-test market models (Silk and Urban 1978). It should be mentioned, that the BTL rule assumes non-negative utilities which are usually received by normalizing estimated individual partworths so that the minimum utility among all feasible combinations receives a utility of 0 and the maximum utility among all feasible combinations receives a utility of 1. Both rules can also be applied to stochastic partworths (e.g., determined via Hierarchical Bayes estimation). Then, p_{ij} is estimated as the mean probability across all draws (see, e.g., Hein et al 2022).

Moreover, in many conjoint analysis applications, a so-called no-choice option is offered to the responding consumers. The introduction of this additional choice alternative makes data collection more realistic in many markets (see, e.g., Vermeulen et al 2008). The estimated utility for this choice alternative allows to predict no-choices of consumers via thresholds (in the deterministic choice rule assumption) or via choice shares (in the probabilistic choice rule assumption).

2.2 Product-line design models

Basing on estimated partworths for attribute-levels and assumed choice rules, product-line design models select attribute-levels of R new products of a focal firm for a status quo market with O own and F foreign established products. Again, x_{jkl} is the indicator whether combination j (J=R+O+F; j=1,...,J) has level l of attribute k (=1) or not (=0). The x_{jkl} values for the first R products are unknown. The case of modifying own established products is included in this formulation by increasing R and decreasing O.

Green and Krieger (1985) distinguish the product-line design problems into buyers' welfare maximization and sellers' welfare maximization depending whether the firm and the consumers "involve the same party" or play "a two-party game": In the first case, the firm designs its new products in order to maximize consumers' utility. In the second case, the consumers are allowed to choose among all available products according to their preferences but the firm designs the attribute-levels of the R new products according to an objective function (e.g., overall revenue, market share, profit) according to its requirements. In the following, we discuss three wide-spread sellers' welfare maximization problems according to this distinction.

2.2.1 Deterministic market share maximization (M_{det})

Market share maximization based on the first-choice rule has often been proposed in product-line design, see, e.g., the approaches by Shocker and Srinivasan (1974); Albers and Brockhoff (1977); Kohli and Krishnamurti (1987); Kohli and Sukumar (1990); Shi et al (2001); Balakrishnan et al (2004); Camm et al (2006); Albritton and McMullen (2007); Vökler et al (2013), as well as many others. The main idea is to maximize the share of consumers that will switch from formerly bought foreign products to one of the new products. Kohli and Krishnamurti (1989) showed that this so-called shares-of-choices problem is NP-hard.

The model formulation M_{det} proposed by Balakrishnan et al (2004) reflects this idea convincingly: Additionally to the R new products (with descriptions to be looked for), O own and F foreign status quo products (with pre-defined descriptions) are under consideration. Estimated partworths β_{ikl} from I consumers are available with respect to K attributes and L_k levels. They are standardized at the individual level—as discussed in the previous section—so that the minimum utility among all feasible combinations receives a utility of 0 and the maximum utility among all feasible combinations receives a utility of 1. Basing on the attribute-level indicators x_{jkl} of the O own and F foreign status products (J=R+O+F; j=R+1,...,J) it is feasible to determine consumers' status quo product based on maximum utility and to concentrate in the following (by rearranging the consumers) on the I' consumers (i=1,...,I') with a foreign status quo product. For those consumers relative partworths are estimated via $\beta^r_{ikl}=\beta_{ikl}-\beta^0_{ikl}$ where β^0_{ikl} reflects the partworths of consumer i's foreign status quo product (i=1,...,I', k=1,...,K, l=1,...,L_k). We now define M_{det}—according to Balakrishnan et al (2004)—as follows:

$$\frac{1}{|I'|}\sum_{i=1}^{|I'|} y_i \longrightarrow \max! \tag{5}$$

subject to

$$\sum_{l=1}^{L_k} x_{jkl} = 1 \quad \forall \quad j = 1,\dots, R;\ k = 1,\dots, K, \tag{6}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} \beta^r_{ikl} x_{jkl} + z_{ij} > 0 \quad \forall \quad i = 1,\dots, I';\ j = 1,\dots, R, \tag{7}$$

$$y_i \le R - \sum_{j=1}^{R} z_{ij} \quad \forall \quad i = 1,\dots, I', \tag{8}$$

$$x_{jkl},\ y_i,\ z_{ij} \in \{0,1\} \quad \forall \quad \begin{matrix} i = 1,\dots, I';\ j = 1,\dots, R; \\ k = 1,\dots, K;\ l = 1,\dots, L_k. \end{matrix} \tag{9}$$

The objective function (5) maximizes the share of consumers that switch from a foreign product to a new product (j=1,...,R). Constraint (6) ensures that each attribute of a new product has exactly one level. Constraint (7) implies z_{ij}=1 if a new product j provides equal or lower utility to consumer i than its current status quo product and z_{ij}=0 otherwise. Constraint (8) ensures y_i=0 if none of the new products provides higher utility to consumer i than its foreign status quo product and y_i=1 otherwise. Constraint (9) forces the decision variables to be binary.

2.2.2 Deterministic profit maximization (P_{det})

Kohli and Sukumar (1990)'s well-known seller's problem are used as an example for profit maximization based on the first-choice rule assumption. Consumer i (i=1,...,I) is assumed to buy the product that maximizes her/his utility among R new as well as O+F own or foreign status quo products. Consumer i buys new product j only if utility u_{ij} exceeds u_i^0, the highest utility among the O+F status quo products. In this case, the relative contribution margin d_{ij}^r—the sum of level-specific relative contribution margins d_{ikl}^r—is generated,

$$d_{ij}^r = \sum_{k=1}^{K} \sum_{l=1}^{L_k} d_{ikl}^r x_{jkl} \quad \forall i = 1,\dots, I;\ j = 1,\dots, R. \tag{10}$$

Please note: d_{ikl}^r is identical to the level-specific contribution margin (d_{ikl}^r=d_{ikl}) if consumer i's status quo product is a foreign status quo product. If consumer i's status quo product is an own status quo product, the level-specific contribution margin of this product (d_{ikl}^0) is subtracted ($d_{ikl}^r = d_{ikl} - d_{ikl}^0$, i=1,...,I, k=1,...,K, l=1,...,L_k). Basing on these notations, our deterministic profit maximization problem can be formalized as follows:

$$\sum_{i=1}^{I} y_i \sum_{j=1}^{R} \sum_{k=1}^{K} \sum_{l=1}^{L_k} d_{ikl}^r x_{ijkl} \longrightarrow \max! \tag{11}$$

subject to

$$\sum_{j=1}^{R} \sum_{l=1}^{L_k} x_{ijkl} = 1 \quad \forall \quad i = 1,\dots, I;\ k = 1,\dots, K, \tag{12}$$

$$\sum_{l=1}^{L_k} x_{ijkl} - \sum_{l=1}^{L_{k'}} x_{ijk'l} = 0 \quad \forall \quad \begin{array}{l} i = 1,\dots, I; j = 1,\dots, R; \\ k, k' = 1, \dots, K; k' < k, \end{array} \tag{13}$$

$$x_{ijkl} + x_{i'jkl'} \le 1 \quad \forall \quad \begin{array}{l} i, i' = 1,\dots, I; i < i';\ j = 1,\dots, R; \\ k = 1,\dots, K; l, l' = 1,\dots, L;\ l < l', \end{array} \tag{14}$$

$$\sum_{j=1}^{R} \sum_{k=1}^{K} \sum_{l=1}^{L_k} \beta_{ikl}(x_{ijkl} - x_{i'jkl}) \ge 0 \quad \forall \quad i, i' = 1,\dots, I;\ i \ne i', \tag{15}$$

$$y_i \sum_{j=1}^{R} \sum_{k=1}^{K} \sum_{l=1}^{L_k} \beta_{ikl} x_{ijkl} \geq y_i(u_{i0} + \epsilon) \quad \forall \quad i = 1,\ldots, I, \tag{16}$$

$$y_i, x_{ijkl} \in \{0, 1\} \quad \forall \quad \begin{array}{l} i = 1,\ldots, I, \;\; j = 1,\ldots, R, \\ k = 1,\ldots, K; \;\; l = 1,\ldots, L_k. \end{array} \tag{17}$$

The objective function (11) maximizes the profit from R new products that exceeds the profit from O own status quo products in competition with F foreign status quo products. The binary decision variables x_{ijkl} (i=1,...,I; j=1,...,R; k=1,...,K; l=1,...,L_k) and y_i (i=1,...,I) have the following meaning: x_{ijkl}=1 (=0) indicates that new product j assigned to consumer i has (does not have) level l of attribute k. y_i=1 (=0) indicates that consumer i selects (does not select) a new product. For these variables, constraint (12) requires that each attribute has only one assigned level per consumer and new product. Constraint (13) requires that across attributes, the level assigned to a consumer corresponds to the same product. Constraint (14) requires that the same level of an attribute must be specified for all consumers assigned to a product. Together, the constraints (12) to (14) result in the requirement that each consumer is assigned one new product. The constraint (15) ensures that a consumer is assigned to the new product that generates her/him maximum utility. That a consumer only buys a new product if it generates higher utility than her/his status quo product is expressed by constraint (16). To ensure that a consumer switches from his status quo product only if the new product has a greater utility than her/his status product, a small constant ϵ is introduced. Finally, the constraint (17) ensures the binary restrictions on the decision variables.

2.2.3 Probabilistic profit maximization (P_{prob})

In this section, we present a profit maximizing model that applies a probabilistic choice rule. Among the proposals by, e.g., Green et al. (1981); Green and Krieger (1992); Choi and DeSarbo (1993, 1994); Gaul et al (1995); Steiner and Hruschka (2000); Chen and Hausman (2000); Steiner and Hruschka (2003); Gaul and Baier (2021), we selected the Gaul et al (1995) model, since it is a flexible model with fixed costs per period at an attribute-level. Again, besides the R new products, O pre-defined own and F foreign status quo products are under consideration. Consumer i's attribute-level contribution margins d_{ikl} and partworths β_{ikl} as well as a weight ω_i (number of products per period bought by consumer i) are used. The periodical fixed costs are estimated by summing up the relevant attribute-level specific periodical fixed costs f_{kl} (k=1,...,K; l=1,...,L_k). This approach also allows to block certain attribute-levels for the new products by defining large f_{kl} values for specific (k,l) combinations. The Gaul et al (1995) model makes use of the BTL choice rule discussed in the last subsection. Due to assumed superiority over the deterministic choice rule and wide spread in test market simulation (Silk and Urban 1978), this flexible alternative to the logit choice rule was and is used for product-line design by many authors (see, e.g., Gaul et al 1995; Schön 2010;

Tsafarakis et al 2011; Vökler and Baier 2020; Gaul and Baier 2021). The model is formulated as follows (see Gaul et al 1995; Gaul and Baier 2021):

$$\sum_{i=1}^{I}\sum_{j=1}^{R+O}\sum_{k=1}^{K}\sum_{l=1}^{L_k}\omega_i d_{ikl}x_{jkl}p_{ij} - \sum_{j=1}^{R}\sum_{k=1}^{K}\sum_{l=1}^{L_k}x_{jkl}f_{kl} \longrightarrow \max! \tag{18}$$

subject to

$$\sum_{l=1}^{L_k}x_{jkl} \le 1 \quad \forall \quad j=1,\ldots,R;\ k=1,\ldots,K, \tag{19}$$

$$p_{ij} = \frac{\left(\sum_{k=1}^{K}\sum_{l=1}^{L_k}\beta_{ikl}x_{jkl}\right)^{\alpha}}{\sum_{j'=1}^{J}\left(\sum_{k=1}^{K}\sum_{l=1}^{L_k}\beta_{ikl}x_{j'kl}\right)^{\alpha}} \quad \forall \quad i=1,\ldots,I;\ j=1,\ldots,R+O, \tag{20}$$

$$\sum_{l=1}^{L_k}x_{jkl} = \sum_{l=1}^{L_{k+1}}x_{j(k+1)l} \quad \forall \quad j=1,\ldots,R;\ k=1,\ldots,K-1, \tag{21}$$

$$x_{jkl} \in \{0,1\} \quad \forall \quad j=1,\ldots,R;\ k=1,\ldots,K;\ l=1,\ldots,L_k, \tag{22}$$

$$p_{ij} \in [0,1] \quad \forall \quad i=1,\ldots,I;\ j=1,\ldots,R+O. \tag{23}$$

The objective function (18) maximizes the weighted contribution margins from the new and the own status quo products minus the fixed costs for the new products. Constraint (19) ensures that a new product has at most one level per attribute. Constraint (20) specifies the BTL choice rule with p_{ij} as the choice/buying probability of consumer i among the new and all status quo products. Constraint (21) ensures that products are offered complete with all attributes. Finally, constraints (22) and (23) reflect the restrictions on the decision variables. Please note that only x_{jkl} values for j=1,...,R are looked for, i.e. descriptions for the new products. The x_{jkl} values for j=R+1,...,J=R+O+F are given, i.e. the description of the own and foreign status quo products.

It should be mentioned that this formulation of a profit maximization problem differs from the one in the last subsection not only with respect to the choice rule and the additional fixed costs but also by allowing identical new products (with respect to the attribute-levels defined) to be introduced. Since—when applying a probabilistic choice rule—these new products (that may be different among each other by attributes not included in the data collection) can generate additional choice probabilities and additional profit, this formulation could have an advantage.

3 One-stage heuristics to solve product-line design problems: an overview

In commercial conjoint analysis applications, the number of attributes and levels (e.g. 3 to 9 attributes, each with 2 to 7 levels according to Selka 2013, see Sect. 1) as well as the derived number of attribute-level combinations (here: up to 7^9=40,353,607) is rather high. Consequently, often, the number of feasible solutions for product-line design problems (the number of feasible sets of, say, 3 to 9 combinations) prevents the application of exact methods for solving in acceptable computing time. Instead, from the introduction of conjoint analysis to marketing (Green and Rao 1969), heuristics have been proposed. Well-known first proposals were developed by Shugan and Balachandran (1977) as well as Zufryden (1977), many others followed. Table 1 gives an overview on recently proposed ones and empirical comparisons.

The overview starts in 1995, since this year reflects the important introduction of Genetic Algorithms (Balakrishnan and Jacob 1995, 1996) to the product-line design literature. Up to now, the proposed method in these references is the only one in Sawtooth Software's Lighthouse system (Orme 2019), the market leader for conjoint analysis software. Another reason for selecting this starting year was a similarly structured former overview on product-line design problems and methods till 1995 by Baier and Gaul (1999).

A closer look at Table 1 reveals that (with few exceptions and in contrast to the proposals in the former overview) mainly one-stage heuristics were proposed and applied since 1995. One-stage heuristics directly search for best sets of attribute-level combinations and omit a foregoing additional stage where promising attribute-level combinations are selected as candidates and the solution space is reduced to sets of these candidates. One exception that refers to two-stage heuristics (especially Green and Krieger's Divide &Conquer, Greedy, and Product Swapping Heuristics from the 1980 s and 1990 s) is Belloni et al (2008a). However, these authors only applied them for comparisons and to small product-line design problems. Also, it can be seen in Table 1, that most proposals contained a comparison with results derived from already known heuristics and/or—in case of smaller problem instances—Complete Enumeration or other methods that guarantee global optimality (e.g. Branch and Bound, Branch and Bound with Lagrange Relaxation). Criteria for the comparisons were accuracy (e.g. "How close was the solution's objective function to the global maximum on average?") and speed (computing time on average). For the comparisons, partworths and status quo products from real-world conjoint analysis applications were used, or—more often—randomly generated. Depending on the number of attributes, levels, and new products to be designed, the number of feasible solutions varies, ranging from 16 to more than 10^{100}. Also, different objective functions, choice rules, and heuristics have been used to define and solve the problems. Moreover, it should be mentioned that the heuristics use different hyperparameters (e.g., selection, crossover, and population maintenance mechanisms in Genetic Algorithms) and parameters (e.g., population percentages and size in Genetic Algorithms) that

Table 1 Solution methods for product-line design problems since 1995: an overview

Reference	Obj. func.	Choi. rule	Sing., mult.	#Feasible solutions	#prob. N, #cons. I, PW &SQP generation	Methods applied (**best methods**)
Balakrishnan and Jacob (1995)	BW, MS	FC	Sing.	256	N=1, PW, SQP: n.a	CE, DPH, **GA**
Nair et al (1995)	BW, MS,P	FC	Mult.	$120–6.3 \cdot 10^5$	N=435, PW: unif., SQP: random	CE, DPH, **BS**
Raman and Chhajed (1995)	P	FC	Mult.	$27–2.1 \cdot 10^{79}$	N=40, I=10, PW: unif., SQP: random	**SA**
Balakrishnan and Jacob (1996)	BW, MS	FC	Sing.	16–390625	N=192, PW: unif., SQP: random	CE, DPH, **GA**
Chen and Hausman (2000)	P	Log.	Mult.	n.a	n.a	CE
Steiner and Hruschka (2000)	P	Log.	Mult	n.a	n.a	CE
Thakur et al (2000)	M	FC	Mult.	120–523776	N=300, PW: unif., SQP: random	GA, **BS**
Alexouda and Paparizzos (2001)	P	FC	Sing., line	$2016–3.66 \cdot 10^{15}$	N=300, PW: unif., SQP: random	**GA**, BS, GA/BS hybrid
Shi et al (2001)	M	FC	Mult.	$3125–5.04 \cdot 10^{102}$	N=217, I=400, PW: beta, unif., SQP: rand	BS, DCH, DPH, GH,GA, NPA, **NPA hybrid**
Tarasewich and McMullen (2001)	M	FC	Sing.	1024–59049	N=120, PW: unif., norm., SQP: random	CE, GA, TS, **pruning**
Steiner and Hruschka (2002)	P	Log.	Mult.	$28–4.6 \cdot 10^{10}$	N=276, PW, SQP: n.a	CE, **GA**, GH
Steiner and Hruschka (2003)	P	Log.	Mult.	$28–4.6 \cdot 10^{10}$	N=276, PW, SQP: n.a	CE, **GA**, GH
Balakrishnan et al (2004)	M	FC	Mult.	$6.12 \cdot 10^{26}–1.88 \cdot 10^{38}$	N=800, I=200, PW: unif., SQP: random	BB, **GA**, BS, GA/BS hybrid
Camm et al (2006)	M	FC	Sing.	$2.95 \cdot 10^5–7.21 \cdot 10^{16}$	N=96, $I \in \{600,1200\}$, PW: beta, SQP: rand	BBLR, GH
Fruchter et al (2006)	P	FC	Sing.	n.a	N=3, $I \in \{30,150,250\}$	GA
Albritton and McMullen (2007)	M	FC	Sing.	$1.0 \cdot 10^5–1.0 \cdot 10^9$	N=4200, $I \in \{200,250, 300\}$, PW: beta	CE, ACO
Belloni et al (2008a)	P	FC	Mult.	$56–5 \cdot 10^{15}$	N=120, $I \in \{50,100\}$, PW: unif., SQP: random	BBLR, BS, CoA, DCH,DPH, GA, NPA, PSH,GH, **SA**
Wang et al (2009)	M	FC	Mult.	$2.49 \cdot 10^{11}–7.95 \cdot 10^{49}$	$I \in \{2,321,1131\}$, PW: beta, SQ:rand	**Branch &Price**

Table 1 (continued)

Reference	Obj. func.	Choi. rule	Sing., mult.	#Feasible solutions	#prob. N, #cons. I, PW &SQP generation	Methods applied (**best methods**)
Tsafarakis et al (2011)	BW, M	BTL	Mult.	$1.58 \cdot 10^{15}$–$3.90 \cdot 10^{67}$	N=160, I ∈{200,700}, PW: unif., SQP: rand	**GA**, PSO
Wang and Curry (2012)	M	FC	Sing.	1,728	Real	BB
Vökler et al (2013)	M	FC	Sing.	625–$1.0 \cdot 10^{20}$	N=450, I ∈{250,500, 1000}, PW: unif	**ACO**, BCO
Bertsimas and Misic (2019)	M, P	FC	Mult.	15504–$2.63 \cdot 10^{23}$	N=480, I ∈{100,200, 500,1000}, rankings	**BBLR**, DCH
Tsafarakis et al (2020)	P	Log.	Mult.	70–$8.06 \cdot 10^{23}$	N=121, I ∈{50,100}, PW: real	**FSTDE**, **GA**, SA
Pantourakis et al (2022)	P	FC	Mult.	70–$8.06 \cdot 10^{23}$	N=121, I ∈{20,75, 100}, PW: real, unif	GA, SA, **CSA**
Tsafarakis et al (2022)	P	FC	Mult.	70–$8.06 \cdot 10^{23}$	N=121, I ∈{50,100}, PW: real, unif	CoA, GA, DCH, DPH, GH, NPA, SA, **TS**
This paper	BW, M, P	FC, BTL	Line	$6.96 \cdot 10^{5}$–$7.33 \cdot 10^{200}$	N=78, I ∈[200,1110], PW: unif., SQP: rand	GA, **CGA**, PSO, **MM**, **MML**, SA

Best methods in each comparison are bolded

BW buyers Welfare; *BTL* BTL choice rule; *#cons.* number of consumers; *FC* first-choice rule; *Log.* Logit choice rule; *M* market share; *Mult.* multiple new products; *n.a.* not available; *P* profit; *PW* partworth; *#prob.* number of problem instances; *Sing.* single new product; *SQP* status quo products; **Exact methods:** Branch and Bound (BB), Branch and Bound with Lagrange Relaxation (BBLR), Complete Enumeration (CE), **One-stage heuristics:** Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), Beam Search (BS), Cluster-based GA (CGA), Coordinate Ascent (CoA), Clonal Selection Algorithm (CSA), Dynamic Programming Heuristic (DPH), Fuzzy Self-Tuning Differential Evolution (FSTDE), Genetic Algorithm (GA), Max-Min Ant System (MM); MM with Local Search (MML), Nested Partition Algorithm (NPA), Particel Swarm Optimization (PSO), Simulated Annealing (SA), Tabu Search (TS); **Two-stage heuristics:** Divide &Conquer Heuristic (DCH), Greedy Heuristic (GH), Product-Swapping Heuristic (PSH) have an influence on accuracy and speed, making conclusions across comparisons and the selection of best heuristics difficult.

We try to overcome these difficulties by extensively comparing four well-known and according to Table 1 advantageous heuristics to our three problems (M_{det}, P_{det}, and P_{prob}) in Sect. 4 (hyperparameter and parameter tuning) and 5 (comparison): Genetic Algorithms (according to Balakrishnan and Jacob 1995, 1996; Steiner and Hruschka 2003), Ant Colony Optimization (Albritton and McMullen 2007), Particle Swarm Optimization (Tsafarakis et al 2011), and Simulated Annealing (Belloni et al 2008a).

4 Selected heuristics and their hyperparameters

In the following, we discuss the four selected one-stage heuristics and apply them to 18 product-line design problem instances. During hyperparameter and parameter tuning, recent improvements from the operations research literature are discussed and tested. Three modifications among these improvements—a new variant of Genetic Algorithms: Cluster-based Genetic Algorithms, and two new variants of Ant Colony Optimization: Max-Min Ant System as well as Max-Min Ant System with Local Search—are later (in Sect. 5) also used in the comparison based on 60 small- to large-size problem instances.

4.1 Problem instances for hyperparameter and parameter tuning

Data for product-line design problems as discussed in the previous sections mainly consist of attribute-level partworths and contribution margins from a sample of consumers as well as attribute-levels of status quo products. In order to develop or compare new solution methods, usually such data is taken from real applications or is generated according to some pre-specified characteristics. So, e.g., Belloni et al (2008a) generated partworths and three status quo products with the number of attributes being 3, 5, or 7, and the number of levels per attribute being 2, 3, 5, or 8. partworths were drawn from iid uniform [0, 1] distributions, status quo products randomly selected. For 3 or 4 new products to be designed, the number of feasible solutions (identical new products not allowed) varied from $\binom{2^3}{3}$=56 to $\binom{8^7}{4}$=8.1 $\cdot$ 10^{23} but was further restricted to problem instances with at most $5 \cdot 10^{15}$ feasible solutions. Belloni et al (2008a) argue that they limited themselves to these small-size problem instances (see Table 1 to compare) since they wanted to check accuracy by calculating the global maximum of the objective functions. They applied an improved variant of Branch and Bound with Lagrange Relaxation that is much faster than Complete Enumeration but is not applicable to problem instances analyzed by Shi et al (2001) and Balakrishnan et al (2004) with up to $5.04 \cdot 10^{102}$ feasible solutions. As a result of their comparison, they found out that—across their rather small problem instances—the Genetic Algorithm almost always (in 99.9% of the cases) and Simulated Annealing always found the global maximum. The computing time of Branch and Bound with Lagrangian Relaxation was on average 659.4 s, of the

Genetic Algorithm 11.8 s, and of Simulated Annealing 131.8 s. The examined two-stage heuristics found the global maximum less frequently but finished much faster, similar quickly to other examined one-stage heuristics (e.g. Beam Search and Nested Partitions Algorithm) with better accuracy but still worse compared to the one-stage heuristics Genetic Algorithm and Simulated Annealing.

In our examination, we want to investigate the performance of our heuristics when confronted to realistic (e.g., larger) product-line design problem instances, where Complete Enumeration and Branch and Bound with Lagrangian Relaxation cannot be applied. Here, in order to deal with realistic problem instances, published characteristics of commercial conjoint analysis applications are helpful. So, e.g., Selka (2013) discusses a sample of all 2,089 conjoint analysis applications performed by a leading market research institute within recent years. Each application is described by its purpose, the branch, the number and types of respondents, the applied conjoint analysis methodology, the number of attributes, the number of levels for each attribute, as well as additional information with respect to reliability and validity of the estimated partworths. On average, the number of attributes was 9.595, ranging from 2 to 51 with a standard deviation of 6.912 and a median of 7. The mean number of levels per attribute was 4.824, ranging from 2 to 47 with a standard deviation of 2.506 and a median of 4. The number of respondents ranged from 23 to 9,801 with a mean of 574.88 and a standard deviation of 469.06. The number of feasible attribute-level combinations ranged from 15 to $9.281 \cdot 10^{31}$ with a median of 2,880 and a mean of $4.448 \cdot 10^{28}$. If 3 to 9 new products are looked for, the number of feasible product-lines ranges from $\binom{15}{3}$=455 to $\binom{9.281 \cdot 10^{31}}{9}$=$1.41 \cdot 10^{282}$ (if identical new products not allowed).

As many of the described product-line design problem instances are rather time consuming to solve, we concentrate in a first step of our investigation—hyperparameter and parameter tuning—on a small set of problem instances with the number of attributes and the number of levels per attribute being 5, 10, or 15 and the number of respondents being 200. If 3 or 9 new products are looked for, $\binom{5^5}{3}$=$5.1 \cdot 10^9$ up to $\binom{15^{15}}{9}$=$1.63 \cdot 10^{153}$ new product-lines are feasible (if identical new products not allowed). These problem instances are much larger (more realistic) than the problem instances analyzed in many former comparisons. Consequently, many of them cannot be solved by Complete Enumeration or Lagrangian Relaxation in acceptable computing time. Instead, they still allow comparisons of many methodological variants with best solutions across all heuristics in reasonable time. We refer in the following to these best solutions across all heuristics for comparisons. Later—in Sect. 5—we extend these comparison to more small- to large-size problem instances, applying fewer but now tuned methodological variants.

The partworths of the 200 respondents as well as the attribute-level contribution margins were generated based on iid uniform distributions and the status quo products were selected randomly, both as proposed by Kohli and Sukumar (1990) as well as Belloni et al (2008a). In contrast to their propositions, however, the partworths were superimposed by multivariate normally distributed dispersion that better reflects today's wide-spread Hierarchical Bayes partworth estimation (see Allenby and Rossi 1998) based on normally distributed a priori and posterior distributional assumptions.

4.2 Genetic algorithms (GA01,...,GA18, CGA)

GAs were introduced to optimization by Holland (1975) and to product-line design by Balakrishnan and Jacob (1995, 1996). They mimic biological evolution by natural selection. Starting point often is a first population with random solutions. A subsequent population emerges by preserving, recombining, and/or mutating selected "best" solutions from the previous one. This generational transition is repeated until a stopping criterion is met. GAs differ in their selection, recombination, and mutation mechanisms during these transitions as well as their initialization and maintenance of the populations. Table 2 gives an overview of GA applications to product-line design. There, the encoding of feasible solutions (the attribute-levels for each new product), the mechanisms used as well as the comparison results are discussed.

It can be seen that especially the hyperparameters selection (truncation or tournament), crossover (1-point or uniform crossover), as well as population maintenance (emigration, Malthusian, modified Malthusian) vary across applications. We will use our problem instances for deciding which mechanisms lead to best solutions and use the following specifications: Selection by truncation was introduced by Holland (1975) and applied to solve product-line design problems by, e.g., Balakrishnan and Jacob (1995, 1996), Belloni et al (2008a), or Tsafarakis et al (2011). A predetermined proportion of "best" solutions is used for preservation, crossover, and/or mutation. Selection by tournament (Goldberg and Deb 1991) is an alternative to truncation. This mechanism was applied by Steiner and Hruschka (2002, 2003) as well as Fruchter et al (2006). Repeatedly, pairs of solutions are randomly selected and the "better" solution of each pair is kept. A third promising alternative for selection, up-to-now not used in product-line design, is stochastic universal sampling (Baker 1987; Goldberg and Deb 1991; Reeves 2003). It selects solutions proportionally to their values of the objective function.

With respect to crossover, 1-point and uniform crossover seem to be wide-spread. Other alternatives, such as the 2- up to 5-point crossover (see, e.g. Balakrishnan and Jacob 1995) were seldomly applied according to Table 2. With 1-point crossover, two solutions swap their attribute-levels starting from a randomly specified attribute. With uniform crossover, the swap is performed across all attributes but only with a defined probability (e.g., 50%). As mechanisms for mutation, the attributewise approach seems to dominate. Each level of a solution is randomly modified with a predefined small probability (the mutation rate). In product-line design, the mutation rate is typically set to 1 divided by the number of attributes and the number of new products (see, e.g., Reeves 2003). Finally, for population maintenance three alternatives to define a new generation with M solutions are used: Emigration preserves M-S solutions from the last generation and allows S new solutions to come from crossover or mutation. Malthusian allows to eliminate preserved solutions when new solutions show higher values of the objective function. Modified Malthusian

Table 2 Applications of Genetic Algorithms (GAs) to product-line design with mechanisms set and results derived

Reference	Enco-ding	Selec-tion	Cross- over	Muta-tion	Pop. size	Pop. maint	Comparison results
Balakrishnan and Jacob (1995)	Binary	Trun- cation	Up to 5-point crossover	Free choice	Free choice; up to 400	Emigr, Malth, mMalth	No systematic comparison
Balakrishnan and Jacob (1996)	Binary	Trun- cation	Uniform crossover	Attr	50,100, 200	Malth	Mutation rate has no influence
Thakur et al (2000)	n.a	Trun- cation	1-point crossover	Attr	20	Emigr	No systematic comparison
Alexouda and Papariz-zos (2001)	Binary	Trun- cation	Uniform crossover	Attr	150	Emigr	No systematic comparison
Shi et al (2001)	n.a	n.a	n.a	n.a	n.a	n.a	20% mutation rate per-forms best
Tarasewich and McMul-len (2001)	n.a	n.a	n.a	n.a	50,100, 200	n.a	Mutation rate has an influence
Steiner and Hruschka (2002)	Binary	Tour- nament	1-point crossover	Attr	30 to 250 by 20	Emigr	Crossover prob. and mutation rates have an influence
Steiner and Hruschka (2003)	Binary	Tour- nament	1-point crossover	Attr	150	n.a	No systematic comparison
Balakrishnan et al (2004)	Multinomial	Trun-cation	Uniform crossover	Attr	400	Emigr	No systematic comparison
Fruchter et al (2006)	Multinomial	Tour- nament	Uniform crossover	Attr	n.a	Emigr	No systematic comparison
Belloni et al (2008a)	n.a	Trun- cation	1-point crossover	5%, attr	500, 2000	mMalth	1-point outperforms uniform
Tsafarakis et al (2011)	Multinomial	Trun- cation	Uniform crossover	4%, attr	100	Malth	No systematic comparison
Tsafarakis et al (2020, 2022)	n.a	Trun- cation	1-point crossover	5%, attr	500, 2000	mMalth	No systematic comparison

Attr attributewise, *emigr* emigration, *Malth* Malthusian, *mMalth* modified Malth, *pop.* population; *maint.* maintenance, *n.a.* not available additionally allows preserved solutions to mutate before their final evaluation (Belloni et al 2008b).

For these hyperparameters under investigation, some additional parameter settings were specified according to recommendations in the literature: (Belloni et al 2008b) proposed for emigration to preserve $M/2$ solutions from one generation to the next. For Malthusian and modified Malthusian, they proposed to use the $M/2$ best product-lines to produce offspring (Belloni et al 2008b). Their $M/2$ parents are carried over into the following generation only if they have a correspondingly higher objective function value after the population is reduced to M. We also follow their proposition to set the population size M to 500 as well as Steiner and Hruschka (2002) to set the discussed crossover probability to 1, since the latter allows better objective function values to be obtained than with a lower crossover probability.

With these specifications of three selection, two crossover, and three population maintenance alternatives and the discussed parameter settings, 3·2·3=18 GA variants (GA01,...,GA18) can be applied to our 18 problem instances, each with 10 runs, with results given in Table 3.

There, for each GA heuristic and each of the three investigated objective functions (M_{det}, P_{det}, and P_{prob} from Sect. 2), the achieved results are reflected by mean values and standard deviations across problem instances and runs. The values of the objective functions are normalized to [0,1] by dividing them by the assumed global maximum (the best value achieved across all four heuristics and their variants in this section). So, from Table 3, it is clear that most heuristics were—on average—able to achieve a P_{prob} solution with a value larger than 99.9% of the assumed global maximum. However, for the other two objective functions and the mean across the three objective functions the results are worse. Only the best performing GA variants (GA05 and GA04) achieve mean maximum (across the runs) values larger than 87% of the assumed global maximum. Moreover, the differences among the mean maximum values are statistically significant according to a non-parametric Friedman test (see, e.g., Carrasco et al 2020) with α=0.05. An additional non-parametric Friedman test with multiple comparisons demonstrates that especially GA variants with truncation for selection or with Malthusian or modified Malthusian for population maintenance significantly outperform GA variants with other mechanisms.

Moreover, since the overall performance of the GA variant with largest mean values across the objective functions (GA05) still was low, the GA and product-line design literature was checked for additional alternatives. So, e.g., some authors tried to improve GA performance by including "best" product-lines from applications of other heuristics into the starting population. However, this approach is known often to lead to local optima. So, e.g., Balakrishnan et al (2004) report no performance improvements by connecting GA and beam search in this so-called "hybrid" manner. Here, we propose an alternative way to augment the initial starting population that relates to Green and Krieger (1987)'s well-known Best-in heuristic that was used by the authors as first stage of a two-stage heuristic. The Best-in heuristic selects respondent-specific utility maximizing attribute-level combinations as candidates. However, in order to restrict the number of candidates, Green and Krieger (1987) propose an iteration across respondents. Individual utility maximizing attribute-level combinations only

Table 3 Mean values (standard deviations) of the objective functions M_{det}, P_{det}, and P_{prob} achieved by applying 18 Genetic Algorithms to 18 problem instances (Largest mean values are in bold)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
GA01 (trun,u emigr)	M_{det}	.6310(.3945)	.8176(.2965)	.5217(.3576)	.7047(.3411)	.0586	2.70(1.32)
	P_{det}	.8218(.2155)		.5933(.3074)		.1465	5.59(.65)
	P_{prob}	.9998(.0004)		.9991(.0015)		.0007	8.37(10.26)
GA02 (trun,u Malth)	M_{det}	.6158(.3910)	.8223(.2962)	.5243(.3608)	.7105(.3396)	.0669	3.91(2.00)
	P_{det}	.8512(.2029)		.6080(.3052)		.1534	8.71(6.58)
	P_{prob}	.9999(.0002)		.9993(.0010)		.0005	11.98(14.44)
GA03 (trun,u mMalth)	M_{det}	.7412(.2412)	.8550(.1893)	.6356(.2728)	.7914(.2353)	.0756	9.50(7.07)
	P_{det}	.8239(.1286)		.7390(.1540)		.0568	29.66(39.39)
	P_{prob}	.9999(.0001)		.9996(.0004)		.0004	14.86(19.81)
GA04 (trun,1 emigr)	M_{det}	.7825(.2014)	.8817(.1577)	.7008(.2174)	.8267(.1921)	.0654	5.81(5.08)
	P_{det}	**.8625(.1064)**		**.7798(.1306)**		.0529	14.99(21.63)
	P_{prob}	.9999(.0002)		.9994(.0008)		.0008	15.78(23.23)
GA05 (trun,1 Malth)	M_{det}	**.7987(.2161)**	**.8828(.1630)**	**.7160(.2222)**	**.8305(.1914)**	.0555	8.37(5.94)
	P_{det}	.8498(.1140)		.7759(.1325)		.0479	22.94(31.68)
	P_{prob}	.9999(.0001)		.9996(.0005)		.0004	24.64(36.69)
GA06 (trun,1 mMalth)	M_{det}	.7680(.2333)	.8642(.1802)	.6673(.2371)	.8041(.2116)	.0642	9.01(6.78)
	P_{det}	.8247(.1260)		.7457(.1398)		.0511	27.53(40.36)
	P_{prob}	.9999(.0002)		.9993(.0008)		.0006	27.27(39.63)
GA07 (tour,u emigr)	M_{det}	.6758(.3333)	.8187(.2569)	.4929(.3136)	.6987(.3154)	.1138	2.92(.96)
	P_{det}	.7806(.1922)		.6053(.2509)		.1535	7.18(4.93)
	P_{prob}	.9996(.0007)		.9980(.0018)		.0018	12.28(17.35)
GA08 (tour,u Malth)	M_{det}	.6168(.3907)	.7828(.3216)	.4968(.3445)	.6803(.3540)	.0872	3.95(1.89)
	P_{det}	.7318(.2968)		.5453(.3312)		.1119	6.41(3.78)
	P_{prob}	.9998(.0004)		.9988(.0019)		.0008	13.79(17.30)
GA09 (tour,u mMalth)	M_{det}	.7245(.2530)	.8521(.1982)	.6193(.2801)	.7857(.2404)	.0684	9.08(5.10)
	P_{det}	.8319(.1329)		.7383(.1471)		.0634	30.85(42.83)
	P_{prob}	.9999(.0001)		.9995(.0006)		.0004	18.02(25.20)
GA10 (tour,1 emigr)	M_{det}	.6982(.2303)	.8170(.2098)	.5966(.2346)	.7480(.2508)	.0694	5.30(3.37)
	P_{det}	.7534(.1725)		.6515(.2046)		.0688	10.69(10.01)
	P_{prob}	.9993(.0012)		.9957(.0046)		.0043	20.44(26.28)
GA11 (tour,1 Malth)	M_{det}	.7743(.1937)	.8708(.1602)	.6860(.2261)	.8158(.2005)	.0607	8.65(6.63)
	P_{det}	.8383(.1177)		.7623(.1334)		.0545	22.66(28.65)
	P_{prob}	.9999(.0002)		.9990(.0009)		.0013	25.79(37.15)
GA12 (tour,1 mMalth)	M_{det}	.7381(.2169)	.8525(.1792)	.6430(.2278)	.7906(.2156)	.0635	8.75(6.49)
	P_{det}	.8196(.1228)		.7295(.1409)		.0594	27.72(38.73)
	P_{prob}	.9999(.0002)		.9994(.0006)		.0006	28.61(41.45)

Table 3 (continued)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
GA13 (stoch,u emigr)	M_{det}	.5661(.2632)	.6402(.2422)	.4627(.2458)	.5507(.2500)	.0669	2.82(1.42)
	P_{det}	.5595(.2545)		.4332(.2397)		.0762	3.46(2.05)
	P_{prob}	.7950(.1033)		.7562(.0971)		.0247	2.24(1.37)
GA14 (stoch,u Malth)	M_{det}	.5335(.3494)	.7286(.3207)	.4462(.3032)	.6400(.3439)	.0628	3.11(1.31)
	P_{det}	.6529(.2733)		.4762(.2706)		.1110	5.59(3.89)
	P_{prob}	.9992(.0013)		.9976(.0031)		.0019	14.22(16.45)
GA15 (stoch,u mMalth)	M_{det}	.7316(.2531)	.8509(.2007)	.6067(.2624)	.7672(.2489)	.0848	9.51(6.29)
	P_{det}	.8213(.1485)		.6954(.1853)		.0954	23.37(27.71)
	P_{prob}	**1.000(.0001)**		**.9996(.0006)**		.0004	17.56(21.59)
GA16 (stoch,1 emigr)	M_{det}	.6240(.2425)	.6669(.2214)	.5140(.2336)	.5819(.2242)	.0688	3.41(1.99)
	P_{det}	.5952(.2469)		.4900(.2269)		.0607	4.25(3.11)
	P_{prob}	.7813(.1082)		.7419(.0984)		.0241	2.11(1.19)
GA17 (stoch,1 Malth)	M_{det}	.7341(.2393)	.8512(.1963)	.6341(.2516)	.7782(.2393)	.0744	7.15(4.39)
	P_{det}	.8198(.1547)		.7052(.1941)		.0779	13.95(11.79)
	P_{prob}	.9997(.0006)		.9953(.0036)		.0069	27.53(36.34)
GA18 (stoch,1 mMalth)	M_{det}	.7325(.0230)	.8470(.1892)	.6411(.2319)	.7855(.2248)	.0633	9.29(6.80)
	P_{det}	.8087(.0136)		.7161(.1676)		.0614	20.12(20.89)
	P_{prob}	.9999(.0002)		.9993(.0006)		.0007	33.75(48.91)

Trun truncation, *tour* tournament, *stoch* stochastic universal sampling, *u* uniform crossover, *1* 1-point crossover, *emigr* emigration, *Malth* Malthusian, *mMalth* modified Malthusian; *PLD* product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); Z_{max}: maximum Z value across 10 runs; $\overline{Z}_{max}$: mean Z_{max} valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{mean}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

become candidates when they provide the current respondent an ϵ 0 utility surplus over the up-to-now selected candidates. In the second stage, candidate sets are evaluated to find "best" profit or market share maximizing product-lines. Even with two-stage heuristics nowadays assumed to be inferior to one-stage heuristics (see, e.g., or the discussion in the previous section Belloni et al 2008a), we use a modified Best-in heuristic to augment our GA starting population. However, we don't rely on individual utility maximizing attribute-level combinations but on segment-specific ones and call the GA heuristic Cluster-based GA (CGA). Also we select more than one candidate per cluster to support variation.

CGA can be easily described in the following way: When searching for R new products, the respondents are clustered via kmeans into R segments based on their partworths. For each of the derived R segments, the J_{best} utility maximizing attribute-level combinations are selected (based on the segment-specific mean partworths). Across the R segments/new products, these J_{best} combinations form J_{best}

Table 4 Mean values (standard deviations) of the objective functions M_{det}, P_{det}, and P_{prob} achieved by applying Genetic Algorithm 5 (GA05 with truncated for selection, 1-point for crossover, and Malthusian for population maintenance) and Cluster-based Genetic Algorithm (CGA) to 18 problem instances (Larger mean values according to a Friedman test with α=0.05 are in bold)

Heur	PLD	$Z_{\max}$	$\overline{Z}_{\max}$	Z_{mean}	$\overline{Z}_{\text{mean}}$	σ_Z	t_{mean}(in s)
GA05 (trun,1 Malth)	M_{det}	.7987(.2161)	.8828(.1630)	.7160(.2222)	.8305(.1914)	.0555	8.37(5.94)
	P_{det}	.8498(.1140)		.7759(.1325)		.0479	22.94(31.68)
	P_{prob}	.9999(.0001)		.9996(.0005)		.0004	24.64(36.69)
CGA (trun,1 Malth)	M_{det}	**.9148(.0750)**	**.9395(.0678)**	**.8497(.0855)**	**.8979(.0949)**	.0441	9.50(10.75)
	P_{det}	.9037(.0532)		.8445(.0664)		.0411	13.26(15.78)
	P_{prob}	.9999(.0002)		.9994(.0007)		.0006	19.19(26.51)

PLD product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); $Z_{\max}$: maximum Z value across 10 runs; $\overline{Z}_{\max}$: mean $Z_{\max}$ valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{\text{mean}}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

initial product-lines that are integrated into the GA starting population (together with random starting solutions). CGA has the advantage that "best" (in a segment-specific utility maximizing manner) and rather diverse starting solutions are used, but within GA they can—in comparison to Green and Krieger (1987)'s two-stage heuristic—be improved. Table 4 compares the new CGA heuristic (with J_{best}=50) to the up-to-now best GA variant (GA05) and demonstrates superiority. The advantage with respect to $\overline{Z}_{\max}$ and $\overline{Z}_{\text{mean}}$ is significant according to a non-parametric Friedman test (see, e.g., Carrasco et al 2020) with α=0.05.

4.3 Particle swarm optimization (PSO)

PSO was originally conceived by Kennedy and Eberhart (1995) for optimizing real-valued functions of real-valued variables. PSO has its roots in artificial life in general and in analyzing the behavior of animal swarms in particular. Algorithmic basis is a population of particles (e.g., artificial birds, fishes) that iteratively adjust their positions in space according to their own past positions and evaluations as well as the past positions and evaluations of other particles (Shi and Eberhart 1998). When it comes to solve optimization problems, the positions in space represents admissible solutions that can be evaluated and improved according to an objective function.

Tsafarakis et al (2011) were the first to apply PSO to product-line design problems. Their objective function was market share based on the BTL choice rule. Positions are defined as vectors of (real) values that reflect whether a specific new product has a specific level of an attribute or not. Since new products only are allowed to be assigned to one level per attribute, the positions must be mapped to adequate indicators before evaluation. Here, e.g., the SPV (Smallest Position Value) mapping implies that for each new product and attribute the level with the lowest value receives a coding of 1 and the others receive a coding of 0.

Tsafarakis et al (2011) showed that this SPV mapping performs best with their product-line design problem instances. Vökler and Baier (2020) supported this hyperparameter selection also for other objective functions with a further sample of product-line design problem instances. Other parameters specified in the literature (e.g., population size: 50, number of iterations: 1,000) also proved in these comparisons to be transferable from general PSO recommendations (Kennedy and Eberhart 1995; Shi and Eberhart 1998).

4.4 Ant colony optimization (ACO, MM, MML)

Solving combinatorial optimization problems using colonies of artificial ants primarily goes back to Dorigo (1992)'s dissertation and reflects the foraging behavior of real ants. Artificial (real) ants repeatedly construct a valid solution (a path to the food) via components (sections) and mark the components with pheromones that reflect the overall solution quality (the shortness of the path). The amount of pheromones at the single components helps other ants to select good components in a probabilistic manner and to construct better solutions (shorter paths). Since its introduction to combinatorial optimization, many extensions and improvements of this algorithmic idea were developed and applied to many NP-hard combinatorial problems, e.g. traveling salesman problems, scheduling problems, and subset problems (see, e.g., Gambardella and Dorigo 1995, 1996; Dorigo et al 1999; Dorigo and Stützle 2003).

Albritton and McMullen (2007) introduced Ant Colony Optimization (ACO) to the product-line design literature. In their approach, similar to PSO, the components are alternative levels for all new products and attributes. They demonstrated that ACO can successfully be applied to small-size problem instances (up to 10^9 feasible solutions). On average, in more than 99.9% of their generated problem instances, ACO was able to find the global maximum of the objective function. Moreover, Vökler et al (2013) applied ACO to even larger product-line design problem instances (up to 10^{20} feasible solutions) and demonstrated superiority over other one-stage heuristics.

Recently, in the ant system literature, Stützle and Hoos (1997)'s modification of the ACO heuristic, the so-called Max-Min Ant System (MM), received remarkable attention. So, e.g., Skinderowicz (2020) describes in his literature review that this heuristic is one of the best-performing ACO variants for solving optimization problem instances. MM differs from standard ACO insofar that (1) only the best ant in each iteration is used for the pheromone update and (2) amounts of pheromone are restricted to maximum and minimum values. These modifications help to avoid early stagnation of the algorithm but still take advantage of the implemented elite strategy. The maximum and minimum amounts of pheromone are defined for each attribute-level so that across all iterations all of them have a chance to be part of the solution. For ACO and MM, the parameter settings of Albritton and McMullen (2007) can be used. Moreover, we introduce a new ACO variant: We combine MM and Green et al. (1989)'s Coordinate Ascent heuristic as a third ACO heuristic. In this MM with Local Search (MML), we additionally

Table 5 Mean values (standard deviations) of the objective functions M_{det}, P_{det}, and P_{prob} achieved by applying Ant Colony Optimization (ACO), Max-Mix Ant System (MM), and Max-Mix Ant System with Local Search (MMS) to 18 problem instances (Groupwise largest mean values according to a Friedman test with multiple comparisons and α=0.05 are in bold)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
ACO	M_{det}	.3386(.2372)	.5162(.3195)	.2872(.2202)	.4704(.3151)	.0290	6.46(6.62)
	P_{det}	.3552(.2539)		.2989(.2256)		.0332	8.96(9.27)
	P_{prob}	.8595(.1001)		.8250(.1022)		.0231	11.32(13.95)
MM	M_{det}	.9171(.1043)	.9314(.0984)	**.8640(.1108)**	.8904(.1206)	.0373	26.13(26.18)
	P_{det}	.8773(.1052)		.8079(.1121)		.0428	27.56(26.40)
	P_{prob}	**.9997(.0009)**		**.9992(.0016)**		.0004	26.65(29.57)
MML	M_{det}	**.9569(.0702)**	**.9645(.0700)**	**.8834(.0846)**	**.9185(.1011)**	.0459	102.55(109.76)
	P_{det}	**.9367(.0902)**		**.8723(.1189)**		.0457	109.32(108.74)
	P_{prob}	**.9999(.0003)**		**.9997(.0005)**		.0002	97.52(101.27)

PLD product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); Z_{max}: maximum Z value across 10 runs; $\overline{Z}_{max}$: mean Z_{max} valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{mean}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

improve the “best” solution by an ant in each iteration (if feasible), by testing all alternative levels of single attributes. Table 5 reflects the results of an application of ACO, MM, and MML to the 18 problem instances within 10 runs.

It is clear that MM and MML outperform ACO. The differences between the mean maximum values and the differences between the mean values are statistically significant according to a non-parametric Friedman test (see, e.g., Carrasco et al 2020) with α=0.05. An additional non-parametric Friedman test with multiple comparisons demonstrates that the pairwise differences are also significant with α=0.05. MML performs best with respect to accuracy. However, this accuracy is achieved by longer computing time.

4.5 Simulated annealing (SAE, SAA)

Simulated Annealing is a popular approach for solving difficult combinatorial optimization problems (Aarts and Korst 1989; Belloni et al 2008a). The method mimics annealing in metallurgy where heating and controlled cooling is used to improve material quality. The approach received its name in the publication by Kirkpatrick et al (1983) who—for the first time—applied this approach to the NP-hard traveling salesman problem. The main idea is to iteratively modify components of an existing solutions in the search for best ones, but to accept during this search with a certain probability deteriorations. The reduction of this probability over time (the cooling process) makes it more and more difficult for worse solutions to be selected and

Table 6 Mean values (standard deviations) of the objective functions M_{det}, P_{det}, and P_{prob} achieved by applying Simulated Annealing with Exponential Cooling (SAE) and Simulated Annealing with Adaptive Cooling (SAA) to 18 problem instances (Larger mean values according to a Friedman test with α=0.05 are in bold)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
SAE	M_{det}	.9824(.0267)	.9551(.0835)	**.9346(.0444)**	.9198(.0845)	.03259	229.60(17.14)
	P_{det}	.9887(.0172)		**.9413(.0344)**		.03256	234.15(19.12)
	P_{prob}	.8943(.1219)		.8835(.1304)		.00642	230.44(20.01)
SAA	M_{det}	.9821(.0257)	.9859(.0235)	.9207(.0437)	.9422(.0534)	.03642	231.34(17.09)
	P_{det}	.9759(.0270)		.9062(.0404)		.04836	235.99(19.01)
	P_{prob}	**.9999(.0003)**		**.9997(.0006)**		.00014	231.55(19.68)

PLD product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); Z_{max}: maximum Z value across 10 runs; $\overline{Z}_{max}$: mean Z_{max} valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{mean}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

ends without early stagnation—hopefully—at the global maximum instead of local ones.

Belloni et al (2008a) were the first to apply this approach to product-line design. Their so-called Simulated Annealing with Exponential Cooling (SAE) uses fixed exponents in the cooling formula. In their comparison, SAE performed best in terms of accuracy. However, Aarts and Korst (1989) criticized that SAE cannot be applied to large-size problem instances. They proposed an approach with a quantum mechanical cooling formula, the so-called Simulated Annealing with Adaptive Cooling (SAA). Additional parameter settings were used according to Aarts and Korst (1989) and Belloni et al (2008a).

Table 6 reflects the results of applying SAE and SAA to our 18 problem instances. Whereas SAA outperforms SAE by higher mean values when averaging the values of three objective functions, this superiority is not significant according to a Friedman test with α=0.05. The superiority comes especially from the P_{prob} product-line design problem (with significant differences) whereas SAE was found significantly superior with respect to the other two problems. Nevertheless we selcted SAA for our comparison in Sect. 5.

5 Empirical comparison

In this section we discuss the application of the six selected one-stage heuristics—Genetic Algorithm (GA05), Cluster-based Genetic Algorithm (CGA), Particle Swarm Optimization (PSO), Max-Min Ant System (MM), Max-Min Ant System with Local Search (MML), and Simulated Annealing with Adaptive Cooling (SAA)—to 60 small- to large-size problem instances across our three objective functions (M_{det}, P_{det}, and P_{prob}) and 10 runs. The hyperparameters and parameters were selected according to the previous section. All calculations were performed on conventional notebooks (Lenovo ThinkPad P53s with an 8th Generation Intel Core i7). All heuristics are implemented in R.

5.1 Problem instances for the empirical comparison

Again (as in the previous section), the sample of 2,089 commercial conjoint analysis applications from Selka (2013) serves as a source. However, in this section, we omit the restriction to few problem instances and use the characteristics of this sample (empirical distribution of the number of respondents, of the number of attributes, of the number of attributes per level) more explicitely. For each investigated problem, we randomly draw the number of attributes K and the number of levels per attribute L_k (k=1,...,K) from the corresponding empirical distributions. Moreover, since the number of respondents I in the sample depends on the number of attribute-levels (the number of individual partworths), a linear model between I and $\sum_{k=1}^{K} L_k$ is estimated ($R^2 = 0.129$) and used to define an adequate number of respondents,

$$I = \lfloor 502.0188 + 2.3968 \cdot \sum_{k=1}^{K} L_k \rfloor. \tag{24}$$

The partworths as well as the attribute-level contribution margins are generated as discussed in the previous section using iid uniform and normal distributions.

As the available commercial conjoint analysis, applications do not provide information on the number of new products looked for, the maximum reported number nine in empirical comparisons was used (Tsafarakis et al 2011, R=9, see, e.g.,][) as a randomly selectable alternative to three new products (R=3). Our comparison assumes no own status products of the focal firm (O=0) but F randomly generated foreign status quo products with F depending on the number of feasible attribute-level combinations defined as:

$$F = \left\lfloor \log_{10} \left[\left(\frac{\prod_{k=1}^{K} L_k}{R} \right) \right] \right\rfloor. \tag{25}$$

So, e.g., for nine attributes (K=9), with four or five levels (e.g., L_1=...=L_4=4, L_5 =...=L_9=5) and nine new products (R=9) we generate randomly eight foreign status quo products (F=8). The dependence of F from the numbers of feasible attribute-level combinations and the number of new products increases coverage and variability of the competition.

Basing on the above discussed empirical distributions and assumptions, 10,000 problem instances were generated with feasible attribute-level combinations ranging from 6 to $7.05 \cdot 10^{34}$ and a median of $6.05 \cdot 10^4$ as well as numbers of feasible solutions (product-lines with R new products) ranging from 56 to $6.15 \cdot 10^{300}$ and a median of $1.95 \cdot 10^{23}$. The number of respondents ranges from 514 to 1,194, the number of foreign status quo products from 1 to 300. If we denote problem instances with less than 10^{41} feasible solutions as small-size problem instances with more than 10^{81} solutions as large-size, and the others as medium-size, we see that the distribution is concentrated on small-size problem instances: Among the 10,000 problem instances we have 73.45% small-size, 19.04% medium-size, and 7.51% large-size problem instances. The thresholds between small-, medium- and large-size problem instances were selected according to Table 1 and to Vökler and Baier (2020) who compared product-line design problem instances from 10^4 up to 10^{41} feasible solutions. As a result of their comparison, they found an overall superior performance (in terms of finding the best solutions across all heuristics) of Genetic Algorithms and Simulated Annealing compared to Particle Swarm Optimization and Ant Colony Optimization, but also that this superiority increased with the problem size across three blocks.

The smallest problem across the three blocks in our comparison looks for a product-line with three new products described by four attributes, of whom three have three and one has six levels. The number of feasible solutions is $\binom{3^3 \cdot 6}{3}=6.96 \cdot 10^5$.

Table 7 Mean values (standard deviations) of the objective functions M_{det}, P_{det}, and P_{prob} achieved by applying Genetic Algorithms (GA), Cluster-based Genetic Algorithms (CGA), Particle Swarm Optimization (PSO), Max-Min Ant System (MM), Max-Min Ant System with Local Search (MMS), and Simulated Annealing (SA) to 60 problem instances (Groupwise largest mean values according to a Friedman test with multiple comparisons and α=0.05 are in bold)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
GA05	M_{det}	.9529(.0927)		.9257(.1065)	.9448(.1029)	.0193	382.36(202.89)
	P_{det}	.9405(.1177)	.9638(.0897)	.9115(.1281)		.0189	394.23(197.45)
	P_{prob}	**.9980(.0086)**		**.9973(.0100)**		.0005	397.80(193.79)
CGA	M_{det}	**.9881(.0169)**	.9855(.0276)	**.9673(.0269)**	**.9698(.0392)**	.0138	395.75(189.93)
	P_{det}	.9727(.0395)		.9484(.0513)		.0164	404.84(186.17)
	P_{prob}	**.9957(.0135)**		.9940(.0156)		.0012	408.33(185.77)
PSO	M_{det}	.6151(.2586)	.6975(.2570)	.5637(.2694)	.6541(.2706)	.0303	245.53(158.89)
	P_{det}	.5843(.2613)		.5300(.2666)		.0317	277.59(168.34)
	P_{prob}	.8930(.0797)		.8685(.0885)		.0161	290.09(175.87)
MM	M_{det}	.9765(.0247)	.9683(.0563)	.9438(.0328)	.9417(.0763)	.0227	304.03(223.42)
	P_{det}	.9284(.0794)		.8814(.0973)		.0311	326.90(226.51)
	P_{prob}	**1.000(.0001)**		**.9998(.0004)**		.0002	312.57(223.63)
MML	M_{det}	**.9925(.0120)**	**.9968(.0081)**	.9590(.0307)	**.9765(.0282)**	.0233	418.74(172.96)
	P_{det}	**.9980(.0052)**		**.9708(.0243)**		.0197	424.03(166.04)
	P_{prob}	**.9998(.0002)**		**.9996**(.0006)		.0002	420.65(171.45)
SAA	M_{det}	.9698(.0357)	.9753(.0418)	.9334(.0535)	.9505(.0616)	.0249	242.43(197.62)
	P_{det}	.9567(.0553)		.9192(.0706)		.0283	251.77(199.13)
	P_{prob}	**.9994(.0028)**		**.9987(.0041)**		.0006	350.02(205.53)

PLD product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); Z_{max}: maximum Z value across 10 runs; $\overline{Z}_{max}$: mean Z_{max} valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{mean}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

The largest problem looks for nine new products described by 36 attributes and has $\binom{2 \cdot 3^8 \cdot 4^{14} \cdot 5^3 \cdot 6^4 \cdot 7^5 \cdot 9}{9}=7.33 \cdot 10^{200}$ feasible solutions. The corresponding number of respondents ranges from 562 to 1,110.

5.2 Results of the empirical comparison

Table 7 reflects the results of the six one-stage heuristics across the three objective functions and the 60 problem instances. All heuristics used hyperparameters and parameters as selected in the previous section. Each combination of heuristic, objective function, and problem was run 10 times. Within each run, additional time restrictions were included. Problem instances of block 1 (small-size problem instances) were terminated after 300 s, problem instances of block 2 (medium-size problem instances) after 450 s, and problem instances of block 3 (large-size problem instances) after 600 s. Best solutions up to this termination were recorded and included in the comparison.

Table 7 clearly shows that across nearly all heuristics (with the exception of PSO), solving profit maximization under a probabilistic choice rule assumption (P_{prob}) leads to similar maximum and mean values across all runs and problem instances. A Friedman test with multiple comparisons and α=0.05 supports this equivalence: The five best heuristics show insignificantly different mean values, but all mean values are significantly different from PSO. Most heuristics need about 400 s per run to find these solutions. If we focus on profit maximization under a deterministic choice rule (P_{det}), MML is significantly superior to the other heuristics, if we focus on market share maximization under a deterministic choice rule (M_{det}), MML and CGA are significantly superior to the other heuristics. Across all three objective functions, the significant superiority of MML also holds.

Table 8 details these differences across the three blocks of problem instances: It can be easily seen, that the Genetic Algorithms GA05 and CGA significantly outperform the other heuristics across small-size problem instances whereas MML significantly outperforms the other heuristics across medium- and large-size problem instances. Striking is the fact, that across the blocks the increase of computing time is acceptable: MML needs on average for small-size problem instances 212.80 s, for medium-size problem instances 450.15 s, and for large-size problem instances 600.46 s. Of course, the time restriction in the three blocks (350, 450, 600 s) limits the computing time considerably, but the overall good performance of the three new heuristics (CGA, MM, MML) is convincing.

Again, we have to mention that the values of the objective functions in Tables 7 and 8 are divided by the objective function value of the best solution found for this problem instance), not the global maximum. So, accuracy can only be judged as a relative accuracy. However, in the first block, we were able to check the solutions of the 15 smallest problem instances by Complete Enumeration on our High Performance Computing Cluster and found that the derived best solutions by our heuristics were indeed—as expected according to former comparisons—global maxima. For the larger problem instances, Complete Enumeration didn't finish within seven days of computing time. For the larger problem instances, currently, we try to implement

Table 8 Mean values (standard deviations) of the objective functions of M_{det}, P_{det}, and P_{prob} achieved by applying Genetic Algorithms (GA), Cluster-based Genetic Algorithms (CGA), Particle Swarm Optimization (PSO), Max-Min Ant System (MM), Max-Min Ant System with Local Search (MMS), and Simulated Annealing (SA) to 60 problem instances (Groupwise largest mean values according to a Friedman test with multiple comparisons and α=0.05 are in bold)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
Block 1 (n=20 small-size problem instances, <10^{41}**feasible solutions):**							
GA05	M_{det}	**.9994(.0029)**	**.9998(.0017)**	**.9952(.0076)**	**.9972(.0061)**	.0040	135.07(82.89)
	P_{det}	**1.000(.0000)**		**.9964(.0068)**		.0029	155.18(97.27)
	P_{prob}	**1.000(.0000)**		**1.000(.0000)**		.0000	175.85(117.93)
CGA	M_{det}	**.9994(.0028)**	**.9997(.0017)**	**.9947(.0084)**	**.9964(.0085)**	.0042	167.27(99.03)
	P_{det}	**.9999(.00045)**		**.9945(.0116)**		.0046	179.82(106.60)
	P_{prob}	**1.000(.0000)**		**1.000(.0000)**		.0000	187.41(116.73)
PSO	M_{det}	.9356(.0746)	.9447(.0838)	.9021(.0871)	.9163(.1021)	.0206	93.89(43.19)
	P_{det}	.9088(.1112)		.8668(.1299)		.0277	112.74(61.99)
	P_{prob}	.9896(.0174)		.9799(.0289)		.0072	112.70(59.34)
MM	M_{det}	.9895(.0172)	.9920(.0158)	.9655(.0223)	.9737(.0296)	.0172	59.34(38.81)
	P_{det}	.9864(.0193)		.9561(.0336)		.0215	67.77(46.40)
	P_{prob}	.9999(.0000)		.9996(.0005)		.0000	62.68(41.80)
MML	M_{det}	.9947(.0105)	.9975(.0075)	.9819(.0188)	.9895(.0183)	.0109	205.61(96.34)
	P_{det}	**.9978(.0072)**		.9868(.0225)		.0108	221.46(94.51)
	P_{prob}	.9999(.0000)		.9996(.0006)		.0002	211.32(101.00)
SAA	M_{det}	.9880(.0194)	.9927(.0151)	.9674(.0271)	.9778(.0270)	.0146	70.95(26.32)
	P_{det}	.9901(.0158)		.9661(.0275)		.0199	75.36(26.66)
	P_{prob}	**1.000(.0000)**		**.9999(.0001)**		.0000	110.66(69.24)
Block 2 (n=20 medium-size problem instances, 10^{41} to 10^{81} feasible solutions):							
GA05	M_{det}	**.9830(.0232)**	.9890(.0184)	**.9536(.0248)**	**.9663(.0326)**	.0209	404.76(53.13)
	P_{det}	**.9839(.0179)**		.9455(.0293)		.0233	419.88(53.06)
	P_{prob}	**1.000(.0000)**		**1.000(.0001)**		.0000	410.36(67.39)
CGA	M_{det}	**.9863(.0120)**	.9885(.0145)	**.9565(.0149)**	.9673(.0278)	.0197	417.32(41.31)
	P_{det}	.9791(.0164)		.9455(.0212)		.0228	431.23(26.60)
	P_{prob}	**1.000(.0002)**		**.9998(.0005)**		.0002	433.91(36.25)
PSO	M_{det}	.5637(.0867)	.6548(.1751)	.5026(.0868)	.5978(.1888)	.0362	213.13(46.73)
	P_{det}	.5179(.0616)		.4442(.0533)		.0390	254.71(77.27)
	P_{prob}	.8827(.0284)		.8465(.0285)		.0226	254.49(43.61)
MM	M_{det}	.9646(.0271)	.9648(.0402)	.9286(.0316)	.9348(.0607)	.0245	283.29(107.08)
	P_{det}	.9300(.0413)		.8761(.0484)		.0344	327.57(114.97)
	P_{prob}	**1.000(.0001)**		.9998(.0003)		.0002	298.74(100.92)

Table 8 (continued)

Heur	PLD	Z_{max}	$\overline{Z}_{max}$	Z_{mean}	$\overline{Z}_{mean}$	σ_Z	t_{mean}(in s)
MML	M_{det}	**.9917(.0114)**	**.9961(.0079)**	**.9523(.0153)**	**.9719(.0235)**	.0263	450.15(.06)
	P_{det}	**.9967(.0053)**		**.9637(.0140)**		.0218	450.16(.07)
	P_{prob}	.9998(.0004)		.9996(.0007)		.0002	450.15(.06)
SAA	M_{det}	.9595(.0324)	.9704(.0382)	.9179(.0462)	.9428(.0569)	.0263	196.21(96.54)
	P_{det}	.9517(.0454)		.9105(.0520)		.0280	203.55(101.34)
	P_{prob}	**1.000(.0000)**		**.9999(.0001)**		.0001	354.38(79.19)
Block 3 (n=20 large-size problem instances, >10^{81} feasible solutions):							
GA05	M_{det}	.8764(.1295)	.9027(.1352)	.8282(.1365)	.8709(.1490)	.0329	607.24(1.95)
	P_{det}	.8377(.1608)		.7925(.1620)		.0305	607.64(1.90)
	P_{prob}	.9941(.0144)		.9920(.0163)		.0014	607.20(1.40)
CGA	M_{det}	.9789(.0224)	.9684(.0397)	**.9502(.0271)**	.9458(.0501)	.0174	602.66(2.23)
	P_{det}	.9390(.0506)		.9051(.0580)		.0218	603.46(1.21)
	P_{prob}	.9872(.0213)		.9821(.0231)		.0035	603.65(.98)
PSO	M_{det}	.3461(.0864)	.4930(.2368)	.2864(.0700)	.4482(.2448)	.0341	429.58(115.68)
	P_{det}	.3261(.1012)		.2790(.0865)		.0285	465.33(109.21)
	P_{prob}	.8068(.0292)		.7792(.0266)		.0186	503.07(91.69)
MM	M_{det}	.9754(.0232)	.9481(.0822)	.9374(.0326)	.9165(.1066)	.0264	569.46(66.24)
	P_{det}	.8689(.1012)		.8121(.1224)		.0375	585.36(54.29)
	P_{prob}	**1.000(.0000)**		**1.000(.0001)**		.0001	576.29(64.97)
MML	M_{det}	**.9910(.0139)**	**.9967(.0090)**	**.9428(.0383)**	**.9681(.0356)**	.0325	600.46(.18)
	P_{det}	**.9994(.0015)**		**.9620 (.0269)**		.0266	600.46(.19)
	P_{prob}	.9998(.0003)		.9995(.0005)		.0002	600.48(.16)
SAA	M_{det}	.9619(.0446)	.9627(.0560)	.9151(.0648)	.9308(.0797)	.0338	460.15(167.80)
	P_{det}	.9282(.0716)		.8809(.0899)		.0370	476.39(154.38)
	P_{prob}	.9982(.0047)		.9964(.0066)		.0016	585.02(40.57)

PLD product-line design problem (Note: Z is the objective function value achieved divided by the objective function value of the best solution found for this problem instance); Z_{max}: maximum Z value across 10 runs; $\overline{Z}_{max}$: mean Z_{max} valus across three objective functions; Z_{mean}: mean Z value across 10 runs; $\overline{Z}_{mean}$: mean Z_{mean} value across the three objective functions; σ_Z: standard deviation of Z across 10 runs; t_{mean}: mean computing time across 10 runs

other exact methods in Gurobi. However, up to now, there are no convincing results with respect to objective values in acceptable computing time.

6 Conclusion and outlook

We presented a comprehensive investigation of selected product-line problem instances and selected one-stage heuristics to solve them. In a first step, four well-known heuristics with alternative hyperparamter and paramter settings—Genetic

Algorithms (GA01,...,GA18), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Simulated Annealing (SAA,SAE)—were applied to a small sample of problem instances. Also, three new variants were developed and firstly applied to product-line design problems: Cluster-based Genetic Algorithm (CGA), Max-Min Ant System (MM), and Max-Min Ant System with Local Search (MML). The six best algorithms among these variants (GA05, CGA, PSO, MM, MML, SAA) then were applied to an additional sample of 60 product-line design problem instances.

Overall, the new algorithms (especially CGA and MML) showed a convincing performance: They found—on average across all problem instances and objective functions—the best maximum and the best mean solutions across 10 runs. Especially MML outperformed all other algorithms significantly based on a Friedman's test with multiple comparisons. However, also G05, MM, and SAA performed quite well, especially when the profit maximization problem with a probabilistic choice rule had to be solved. The weakest algorithm was—again as in Vökler and Baier (2020)—PSO. The superiority of MML is even more striking when it comes to medium- or large-size problem instances (with 10^{41} or more solutions). Here, MML clearly outperforms the other algorithms.

The derived results are important, also since in this paper—for the first time when analyzing product-line design problems—a large (n=2,089) sample of commercial conjoint analysis applications formed the basis for deriving results. The number of attributes, the number of levels per attribute, and the number of respondents were selected from their empirical distributions in the sample. This allows to discuss the performance of the algorithms in a real-world setting. Consequently, the generated problem instances in our comparison are much larger than the ones used in former comparisons (up to $7.33 \cdot 10^{200}$ feasible solutions). Against this background, also the results with respect to computing time are promising: In at most 600 s, conventional laptops are nowadays able to provide solutions for such large problem instances.

Our study has limitations. So, e.g., the four (in Sect. 4) respectively six (in Sect. 5) selected heuristics do not include all recently published proposals of one-stage heuristics with promising results. Table 1 already contains interesting alternatives that should be further developed and investigated: Bertsimas and Misic (2019) proposed an interesting exact mixed integer optimization algorithm for product-line design, applied up to now to at most 3,584 feasible attribute-level combinations (computing time: 606.22 s). Tsafarakis et al (2020, 2022) and Pantourakis et al (2022) introduced new promising one-stage heuristics to the product-line problem: The so-called Fuzzy Self-Tuning Differential Evolution, Tabu Search, and the Clonal Selection Algorithm. All of them demonstrated (slight) superiority over Genetic Algorithms and Simulated Annealing. However, their applications seem currently restricted to small-size problem instances similar to Belloni et al (2008a) with up to 8^7=2,097,152 feasible attribute-level combinations and $8.06 \cdot 10^{23}$ feasible solutions. Other developments are also promising, e.g., in assortment planning (see Sect. 1) and could be applied to product-line design problems in the future. However, up to now, the restriction to small-size problem instances in assortment planning is even stronger, only allowing up to 1,000 candidates instead of the needed large number of attribute-level combinations. Moreover, the hyperparameter and parameter tuning in Sect. 4 could—of course—be much more elaborated, even though it was already performed with a considerable amount of work and coverage.

Another shortcoming of this paper is the restriction in accuracy checking to the best solutions across all heuristics. This limitation is again connected to the real-world problem instance sizes where Complete Enumeration or Lagrangian Relaxation is not feasible in acceptable computing time (for us: less than seven days per run on a conventional notebook). The objective function value of the best solution found for this problem instance was also used by many other authors for these reasons see, e.g., Shi et al (2001); Balakrishnan et al (2004), but is—of course—not completely convincing. However, former comparisons with small-size problem instances showed that many one-stage heuristics are able to find the global maximum and—as we use a large set of diverse heuristics—we assume that the probability of detecting the global maxima by at least one heuristic is high. Moreover, from a practical point of view, a manager must and will be satisfied with a best solution among all available applications of algorithms.

Also, the concentration on the three discussed objective functions is a limitation, even so they were selected according to their usage in other comparisons. They assume linear-additive utilities (wide-spread according to Wittink and Cattin 1989; Orme 2019). However, in many commercial conjoint analysis applications, interaction effects (e.g., between a price attribute and a brand attribute) are of high importance and must be considered. The discussed problems and heuristics can be adapted, but—of course—effects of these adaptation to the results of our comparison are not discussed here. Moreover, many commercial conjoint analysis applications have real-valued attributes (e.g. prices, weights, costs). These attributes can be converted to nominal attributes before solving the product-line design problem. However, some product-line design problems exlicitely allow to deal with real-valued attributes (e.g. Dobson and Kalish 1988, 1993). Our comparison cannot answer the question which heuristics are better suited to solve such problems.

The managerial implications of this study are manifold: We could show that one-stage heuristics are able to solve large product-line problem instances and that some heuristics significantly perform better than others. The new one-stage heuristics—CGA, ML, and MML—outperform the wide-spread standard one in this field: the Genetic Algorithm proposed by Balakrishnan and Jacob (1995, 1996) that is implemented in Sawtooth Software's Lighthouse system (Orme 2019). The underlying R code of our heuristics is available for interested researchers but is—of course—also easy to be implemented based on this paper and related references. Moreover, since the heuristics are so fast with a convincing accuracy, they can also be used to flexibly predict performance in a dynamic market comparable to a real options approach: In a first step, a product-line could be designed based on the estimated partworths and established status quo products. Then, in a second step, expected changes of partworths and of status quo products could be specified and the robustness of the product-line against these changes could be checked. Since conjoint analysis applications are wide-spread and highly accepted for product and brand management (see Sect. 2), such additional evaluations in dynamic markets could be of high relevance.

Acknowledgements The authors thank the editor, the guest editors, and the two reviewers very much for their valuable suggestions for clarifications and improvements. Our investigation and further research has benefited a lot from their critical reflection.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The data that support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interests The authors are researchers in the field of conjoint analysis and product-line design but have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Aarts E, Korst J (1989) Simulated annealing and Boltzmann machines. Wiley, New York, NY

Albers S, Brockhoff K (1977) A procedure for new positioning in an attribute space. Eur J Oper Res 1(4):230–238

Albritton MD, McMullen PR (2007) Optimal product design using a colony of virtual ants. Eur J Oper Res 176(1):498–520. https://doi.org/10.1016/j.ejor.2005.06.042

Alexouda G, Paparizzos K (2001) A genetic algorithm approach to the product line design problem using the seller's return criterion: An extensive comparative computional study. Eur J Oper Res 134(1):165–178

Allenby GM, Rossi PE (1998) Marketing models of consumer heterogeneity. J Econom 89(1–2):57–78

Baier D, Gaul W (1999) Optimal product positioning based on paired comparison data. J Econom 89(1–2):365–392

Baier D, Kurz P (2021) Conjointanalyse: verbreitung und validität kommerzieller anwendungen im zeitvergleich. In: Baier D, Brusch M (eds) Conjointanalyse: methoden - anwendungen - praxisbeispiele, 2nd edn. Springer, Berlin Heidelberg, Deutschland, pp 35–49

Baier D, Polasek W (2003) Market simulation using Bayesian procedures in conjoint analysis. In: Schwaiger M, Opitz O (eds) Exploratory data analysis in empirical research. Springer, Berlin, pp 413–421

Baker JE (1987) Reducing bias and inefficiency in the selection algorithm. In: Grefenstette JJ (ed) Proceedings of the 2nd International Conference on Genetic Algorithms, vol 206. L. Erlbaum Associates Inc., Hillsdale, NJ, pp 14–21

Balakrishnan PV, Jacob VS (1995) Triangulation in decision support systems: algorithms for product design. Decis Support Syst 14(4):313–327

Balakrishnan PV, Jacob VS (1996) Genetic algorithms for product design. Manage Sci 42(8):1105–1117

Balakrishnan PV, Gupta R, Jacob VS (2004) Development of hybrid genetic algorithms for product line designs. IEEE Trans Syst Man Cybern 34(1):468–483

Belloni A, Freund R, Selove M et al (2008) Optimizing product line designs: Efficient methods and comparisons. Manage Sci 54(9):1544–1552. https://doi.org/10.1287/mnsc.1080.0864

Belloni A, Freund R, Selove M et al (2008) Optimizing product line designs: efficient methods and comparisons: an electronic companion. Manag Sci. https://doi.org/10.1287/mnsc.1080.0864

Bertsimas D, Misic V (2019) Exact first-choice product line optimization. Oper Res 67(3):651–670
Bradley R, Terry M (1952) Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika 39:324–345
Camm JD, Cochran JJ, Curry DJ et al (2006) Conjoint optimization: an exact branch-and-bound algorithm for the share-of-choice problem. Manage Sci 13:435–447
Carrasco J, García S, Rueda M et al (2020) Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review. Swarm Evol Comput 54(100):665
Chen KD, Hausman WH (2000) Technical note: mathematical properties of the optimal product line selection problem using choice-based conjoint analysis. Manage Sci 46(2):327–332
Choi CS, DeSarbo WS (1993) Game theoretic derivations of competitive strategies in conjoint analysis. Mark Lett 4(4):337–348
Choi CS, DeSarbo WS (1994) A conjoint-based product designing procedure incorporating price competition. J Prod Innov Manag 11(5):451–459
Debreu G (1960) Review of rd luce, individual choice behavior: a theoretical analysis. Am Econ Rev 50(1):186–188
Désir A, Goyal V, Zhang J (2021) Technical note-capacitated assortment optimization: hardness and approximation. Oper Res 70(2):893–904
Dobson G, Kalish S (1988) Positioning and pricing a product line. Mark Sci 7(2):107–125
Dobson G, Kalish S (1993) Heuristics for pricing and positioning a product-line using conjoint and cost data. Manage Sci 39(2):160–175
Dorigo M (1992) Optimization, learning and natural algorithms. In: PhD Thesis, Politecnico di Milano, Dipartimento di Elettronica, Italia
Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover F, Kochenberger GA (eds) Handbook of metaheuristics. Springer, Boston, MA, pp 250–285
Dorigo M, Di Caro G, Gambardella L (1999) Ant algorithms for discrete optimization. Artif Life 5(2):137–172
Fruchter G, Fligler A, Winer R (2006) Optimal product line design: genetic algorithm approach to mitigate cannibalization. J Optim Theory Appl 131(2):227–244
Gambardella L, Dorigo M (1995) Ant-q: a reinforcement learning approach to the traveling salesman problem. Machine learning proceedings 1995. Kaufmann, Morgan, Burlington, MA, pp 252–260
Gambardella L, Dorigo M (1996) Solving symmetric and asymmetric tsps by ant colonies. In: Proceedings of the IEEE International Conference on Evolutionary Computation. IEEE International Conference on Evolutionary Computation, Nagoya, Japan, pp 622–627
Gaul W, Baier D (2021) Simulation und optimierung auf basis der conjointanalyse. In: Baier D, Brusch M (eds) Conjointanalyse: methoden - anwendungen-praxisbeispiele, 2nd edn. Springer, Berlin Heidelberg, Deutschland, pp 275–306
Gaul W, Aust E, Baier D (1995) Gewinnorientierte Produktliniengestaltung unter Berücksichtigung des Kundennutzens. Zeitschrift für Betriebswirtschaftslehre 65(8):835–855
Goldberg DE, Deb K (1991) A comparative analysis of selection schemes used in genetic algorithms. Found Gen Algorithms 1:69–93
Green PE, Rao VR (1969) Nonmetric approaches to multivariate analysis in marketing. In: Working paper, Wharton School, University of Pennsylvania, Philadelphia, PA
Green P, Krieger A (1985) Models and heuristics for product line selection. Mark Sci 4(1):1–19
Green P, Krieger A (1987) A consumer-based approach to designing product line extensions. J Prod Innov Manag 4:21–32
Green PE, Krieger AM (1988) Choice rules and sensitivity analysis in conjoint simulators. J Acad Mark Sci 16(1):114–127
Green PE, Krieger AM (1989) Recent contributions to optimal product positionig and buyer segmentation. Eur J Oper Res 4(1):127–141
Green PE, Krieger AM (1992) An application of a product positioning model to pharmaceutical products. Mark Sci 2(1):117–132
Green PE, Carroll JD, Goldberg SM (1981) A general approach to product design optimization via conjoint analysis. J Mark Res 45(3):17–37
Green PE, Krieger AM, Zelnio RN (1989) A componential segmentation model with optimal product design features. Decis Sci 20(2):221–238
Hein M, Goeken N, Kurz P et al (2022) Using Hierarchical Bayes draws for improving shares of choice predictions in conjoint simulations: A study based on conjoint choice data. Eur J Oper Res 297(2):630–651

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MA
Kennedy J, Eberhart R (1995) Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks IV, Perth, Australia, pp 1942-1948
Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. science 220(4598):671–680
Kohli R, Krishnamurti R (1987) A heuristic approach to product design. Manage Sci 33(12):1523–1533
Kohli R, Krishnamurti R (1989) Optimal product design using conjoint analysis: computational complexity and algorithms. J Oper Res 40(2):186–195
Kohli R, Sukumar R (1990) Heuristics for product-line design using conjoint analysis. Manage Sci 36(12):1464–1478
Louviere J (1988) Conjoint analysis modelling of stated preferences. JTEP 22(1):93–119
Luce R (1959) Individual choice behavior-a theoretical analysis. Wiley & Sons, New York
Luce R, Tuckey J (1964) Simultaneous conjoint measurement: a new type of fundamental measuring. J Math Psychol 1:1–27
McFadden D (1976) Quantal choice analysis: a survey. Ann Econ Soc Meas 5:363–90
Nair SK, Thakur LS, Wen KW (1995) Near optimal solutions for product line design and selection: Beam search heuristics. Manage Sci 41(5):767–785
Orme B (2019) Getting started with conjoint analysis, 4th edn. Research Publishers LLC, Manhattan Beach, CA
Pantourakis M, Tsafarakis S, Zervoudakis K et al (2022) Clonal selection algorithms for optimal product line design: a comparative study. Eur J Oper Res 298(2):585–595
Punj G, Staelin R (1978) The choice process for graduate business schools. J Mark Res 15:588–598
Raman N, Chhajed D (1995) Simultaneous determination of product attributes and prices, and pproducti prozess in product-line design. J Oper Manag 12(3–4):187–204
Reeves C (2003) Genetic algorithms. In: Kochenberger G (ed) Glover F. Kluwer Academic Publishers, Norwell, MA, Handbook of Metaheuristics. International Series in Operations Research & Management Science
Roberts JH, Kayande U, Stremersch S (2014) From academic research to marketing practice: exploring the marketing science value chain. Int J Res Mark 31(2):144–146
Rusmevichientong P, Shmoys D, Tong C et al (2014) Assortment optimization under the multinomial logit model with random choice parameters. Prod Oper Manag 23(11):2023–2039
Schön C (2010) On the optimal product selection problem with price discrimination. Manage Sci 56(5):896–902
Selka S (2013) Validität computergestützer Verfahren der Präferenzmessung. Dr, Kovac, Hamburg, Germany
Selka S, Baier D (2014) Kommerzielle anwendung auswahlbasierter verfahren der conjointanalyse: eine empirische untersuchung zur validitätsentwicklung. Mark ZFP 36(1):54–64
Selka S, Baier D, Kurz P (2014) The validity of conjoint analysis: An investigation of commercial studies over time. Data Anal Mach Lear Knowl Discov. Springer, pp 227–234
Sen A, Atamtürk A, Kaminsky P (2018) A conic integer optimization approach to the constrained assortment problem under the mixed multinomial logit model. Oper Res 66(4):994–1003
Shi L, Ólafsson S, Chen Q (2001) An optimization framework for product design. Manage Sci 47(12):1681–1692
Shi Y, Eberhart R (1998) A modified Particle Swarm Optimizer. In: IEEE International Conference on Evolutionary Computation Proceedings, pp 69–73
Shocker AD, Srinivasan V (1974) A consumer-based methodology for the identification of new product ideas. Manage Sci 20(6):921–937
Shugan S, Balachandran V (1977) A mathematical programming model for optimal product line structuring. In: Working paper, Series 7734 (October), Graduate School of Business, University of Rochester, Rochester, NY
Silk AJ, Urban GL (1978) Pre-test-market evaluation of new packaged goods: A model and measurement methodology. J Mark Res 15(2):171–191
Skinderowicz R (2020) Implementing a GPU-based parallel MAX-MIN ant system. Futur Gener Comput Syst 106:277–295
Steiner W, Hruschka H (2000) Conjoint-basierte Produkt(linien)gestaltung unter Berücksichtigung von Konkurrenzreaktionen. OR Spektrum 22(1):71–95
Steiner W, Hruschka H (2002) Produktliniengestaltung mit genetischen Algorithmen. Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung 54(7):575–601

Steiner W, Hruschka H (2003) Genetic algorithms for product design: how well do they really work? Int J Mark Res 45(2):229–240
Stützle T, Hoos H (1997) MAX-MIN ant system and local search for the traveling salesman problem. In: Proceedings of 1997 IEEE International Conference on Evolutionary Computation, pp 309–314
Tarasewich P, McMullen PR (2001) A pruning heuristic for use with multisource product design. Eur J Oper Res 128(1):58–73
Thakur L, Nair S, Wen KW et al (2000) A new model and solution method for product line design with pricing. J Oper Res Soc 51(1):90–101
Tsafarakis S, Marinakis Y, Matsatsinis N (2011) Particle swarm optimization for optimal product line design. Int J Res Mark 28(1):13–22. https://doi.org/10.1016/j.ijresmar.2010.05.002
Tsafarakis S, Zervoudakis K, Andronikidis A et al (2020) Fuzzy self-tuning differential evolution for optimal product line design. Eur J Oper Res 287(3):1161–1169. https://doi.org/10.1016/j.ejor.2020.05.018
Tsafarakis S, Zervoudakis K, Andronikidis A (2022) Optimal product line design using tabu search. J Oper Res Soc 73(9):1–12
Vermeulen B, Goos P, Vandebroek M (2008) Models and optimal designs for conjoint choice experiments including a no-choice option. Int J Res Mark 25(2):94–103
Vökler S, Krausche D, Baier D (2013) Product design optimization using ant colony and bee algorithms: a comparison. In: Lausen B, Van den Poel D, Ultsch A (eds) Algorithms from and for nature and life. Springer, Cham, Deutschland, pp 491–498
Vökler S, Baier D (2020) Investigating machine learning techniques for solving product-line optimization problems. Arch Data Sci Series A 6(1–11)
Wang X, Curry DJ (2012) A robust approach to the share-of-choice product design problem. Omega 40(6):818–826
Wang X, Camm JD, Curry DJ (2009) A branch-and-price approach to the share-of-choice product line design problem. Manage Sci 55(10):1718–1728
Wittink D, Cattin P (1989) Commercial use of conjoint analysis: an update. J Mark 53(3):91–96
Zufryden FS (1977) A conjoint measurement-based approach for optimal new product design and market segmentation. In: Shocker A (ed) Analytic approaches to product and market planning. Marketing Science Institute, Cambridge, MA, pp 100–114

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.