CHALMERS | UNIVERSITY OF GOTHENBURG

PREPRINT 2011:15

Pure Categorical Optimization – a Global Descent Approach

PETER LINDROTH
MICHAEL PATRIKSSON

Department of Mathematical Sciences
Division of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg Sweden 2011

Preprint 2011:15

Pure Categorical Optimization – a Global Descent Approach

Peter Lindroth, Michael Patriksson

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg, Sweden

Gothenburg, May 2011

Preprint 2011:15
ISSN 1652-9715

Matematiska vetenskaper
Göteborg 2011

Pure Categorical Optimization – a Global Descent Approach

Peter Lindroth[*][†] Michael Patriksson[†]

Abstract

In this article we introduce and study the *pure categorical optimization problem*. This is a problem in discrete variables, but where the discrete variables have no natural ordering in the decision space. It is argued that such a problem is a natural framework for the study of, e.g., the problem of finding the right configuration for a customer for certain types of platform-based products.

It is shown that the problem can be reformulated into several different nonlinear integer programming models that all are equivalent from a categorical point of view. We investigate its mathematical properties; in particular we establish properties corresponding to those of continuity and convexity for numerical optimization problems.

For the solution of the problems we propose to extend the discrete global descent method from the area of (numerical) nonlinear integer programming. We suggest extensions of the principal algorithmic steps within these methods in order to adapt it to categorical problems, utilizing the fact that there are many equivalent problem formulations.

Numerical results are provided when applying the proposed methods both to some standard test problems from the literature, and to real-world problems concerning the configuration of heavy-duty trucks. It is concluded that problems in this class can be solved using mathematical techniques. In particular, it is possible to utilize the discrete global descent method, and it is concluded that the extensions added in order to adapt it to categorical problems also improve its performance.

Keywords: pure categorical optimization, categorical variables, discrete global optimization, global descent methods, nonlinear integer programming, configuration

1 Introduction and motivation

A *pure categorical optimization problem* is a discrete non-linear optimization problem where all variables can be given a numerical value, but where the values themselves

[*]Volvo 3P, Chassis & Vehicle Dynamics, Chassis Strategies & Vehicle Analysis, SE-405 08, Gothenburg, Sweden. E-mail: peter.lindroth@volvo.com

[†]Mathematical Sciences, Chalmers University of Technology, and Mathematical Sciences, University of Gothenburg, SE-412 96 Gothenburg, Sweden. E-mail: {peter.lindroth, mipat}@chalmers.se

do not have a physical meaning as they simply enumerate the possible variable realizations. The following example illustrates a real-world problem belonging to this class, and serves to motivate the study of such problems and their solution.

EXAMPLE 1.1 *Volvo 3P is the business unit within the Volvo group responsible for product development for the truck brands within the Volvo group. Their strategy is to create modularized parts with common interfaces such that a large number of truck configurations matching customer needs at different markets can be built using different combinations of the parts; there are in fact very few completely identical trucks that are built. A truck is specified using about 500 different options, each to be selected from a finite set of alternatives. One example is the color of the cab, another is the type of suspension to be used for the rear axle. In general, there is no natural ordering of the alternatives for each option. Given a customer on a specific market and with a specific transport mission, the task is to find the truck specification best matching his/her needs.*

The concepts *Product family design* and *Platform-based product development* have received lots of attention during the last decades as a way of handling the increasingly competitive market situation, in which it is required to meet customized demands while at the same time keeping development and manufacturing costs down ([23, 39]). The idea is to use shared technology with parts or modules that are used in different combinations and / or sizes in products targeted at specific market niches. Research within these areas involves many aspects ranging from soft topics such as how to create an efficient organization around the development of technology that is to be shared within a product, and hence among different design groups (cf., e.g., [20, 21]), to quantitative optimization in which particular design problems are modelled and solved as mathematical programming problems. Key references for the latter topic include [9, 10, 24, 37, 40]; see also the references in [39], in which it is concluded that there is no standard optimization technique for product family design problems because of the principal differences among them.

The problem illustrated in Example 1.1 is a special case of a product family design problem: there is a fixed number of modules or components to select from and the variables representing these selections are unorderable placeholders for components in a list. In engineering design terms, the problem could be denoted as a *fixed configuration problem* ([7]) (the structure of a product is defined and what is left is to select a module for each dimension of the structure), a *component selection problem* ([8, 9]) or a *catalog-based design problem* ([10]). The full problem for a company like Volvo 3P amounts to finding the components such that the feasible combinations (that is, configurations) match the varying customer demands on the different markets where the trucks are sold. This problem can be formulated as a multi-objective categorical optimization problem. Example 1.1 is a restriction of this problem to that of finding the right truck given an already defined product structure and given a specific customer and his / her specific needs, i.e., his / her individual (scalar-valued) utility function. This problem is naturally associated with the sales process when searching for the right truck for a given customer.

The primary purpose of this article is to argue that problems of the type illustrated in Example 1.1, denoted as *pure categorical optimization problems*, could be analyzed and solved using mathematical techniques. A secondary purpose is to introduce a particular solution method based on a recently developed tool for nonlinear discrete optimization.

During the last decade, a number of papers concerning *mixed-variable programming* (or MVP) have started to appear in the optimization literature (cf. [2, 3, 4, 5, 27, 41, 42]). In these problems, the variable types are allowed to be continuous, discrete or categorical. Examples of applications of MVPs include surface structure determination for nanomaterials ([46]), optimal sensor placement ([6]) and the design of thermal insulation systems ([25]). Important to note is that the term *mixed-variable programming* is not reserved for problems containing categorical variables, but is also sometimes used to state that a problem contains both continuous and (numerically) discrete variables (cf., e.g., [26]). A typical use of a categorical variable could be the material selection of a component in a mechanical construction. The continuous variables then represent the dimensions of the components, and the numerically discrete variables are, e.g., dimensions that have to be selected from a specified list, or a number of layers in the design of some material. In the applications presented, most variables are continuous or numerically discrete; only a few are of a categorical type. We consider a problem in which all variables are categorical, making the necessary theoretical and algorithmical development significantly different.

Important to note is that the term *mixed-variable programming* is not reserved for problems containing categorical variables, but is also sometimes used to state that a problem contains both continuous and (numerically) discrete variables (cf., e.g., [26]).

Is is noted in the recent articles [14, 15] that a common feature of real-life design optimization problems is that choices can only be made from some specified list of alternatives. A natural way of modeling such alternatives is by using categorical, or, as they are denoted in these articles, *integer choice variables*. It is assumed that a certain alternative represents a set of values for a number of parameters, i.e., a point in a multi-dimensional space, and also that the transformation to the multi-dimensional space is known. Using these transformations, the authors suggest a branching technique for the integer choice variables by considering the corresponding points in the multi-dimensional parameter space, implying the existence of an implicit distance measure between the categorical variables. With this branching technique, it is possible to use branch-and-bound methods for mixed-integer nonlinear programming to solve the problems. This method is however not applicable to the most general pure categorical programming problem, since even if a categorical variable corresponds to certain values of a set of parameters in, say, a simulation model, the transformation might depend on the values of other decision variables or even be unknown.

To conclude, we have not found anyone studying the general pure categorical problem from a mathematical perspective which motivates that we do it. Pure categorical optimization problems have to be solved, as demonstrated by the problem in Example 1.1.

The article is structured as follows. We begin in Section 2 by formulating the pure categorical optimization problem in mathematical terms, and note that there is a family of equivalent nonlinear integer programs corresponding to the categorical problem. We also show why categorical variables must be considered for some applications, since methods based on relaxation and convex combinations of the categorical variables will in general not work.

In Section 3 we define local optimality in a categorical context and relate it to local optimality in the numerically discrete world. We show the equivalence of the problems in the family of nonlinear integer programs corresponding to the categorical problems. We then suggest some mathematical properties for functions defined over categorical spaces related to continuity and convexity for numerical problems.

To illustrate that mathematical methods can be used for solving pure categorical problems, we develop in Section 4 a simple neighborhood search method to find local categorical minimizers. We then present the discrete global descent method for numerically discrete optimization. We suggest extensions to this method in order to adapt it to categorical problems, making use of the whole family of equivalent problem representations. The method suggested is guaranteed to converge to a local minimum, and by using the global descent approach, the hope is that the local minimum is also a good local minimum.

Some experimental results are shown in Section 5 when applying the method both to standard test examples from the literature and to real-world problems concerning the configuration of trucks. We end in Section 6 with an outlook on possible future work.

2 Problem formulation

We denote a solution to a categorical optimization problem by a *configuration*, inspired by Example 1.1. A general pure categorical problem of minimizing a real-valued function over a feasible set of configurations can be written as that to

$$\begin{aligned} &\text{minimize} && \tilde{f}(\tilde{\mathbf{x}}), \\ &\text{subject to} && \tilde{\mathbf{x}} \in \Omega, \end{aligned} \tag{1}$$

where $\tilde{f} : \Omega \to \mathbb{R}$, and where Ω is assumed to be a finite set of feasible configurations. For a real-world problem, the size of the feasible decision space might be too large to allow for a brute-force calculation of all the feasible alternatives, wherefore more sophisticated procedures should be developed.

Observe that with some arbitrary ordering of the variables, a categorical optimization problem can be formulated as that to

$$\begin{aligned} &\text{minimize} && \hat{f}(\mathbf{x}), \\ &\text{subject to} && g_j(\mathbf{x}) \leq 0, \quad j \in \mathcal{J}, \\ &&& \mathbf{x} \in X \subseteq \mathbb{Z}^n. \end{aligned} \tag{2}$$

where $\hat{f} : \{\mathbf{x} \in X \mid g_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\} \to \mathbb{R}$ is a reformulation of $\tilde{f}$ to a function of the (physically meaningless) numbers of the variables counting their list positions. We call this transformation from (1) to (2) (or (4)) to *numerize* the problem. Without any loss of generality we let $X = \prod_{i=1}^{n} X^i$, where $X^i = \{1, \ldots, m_i\}$ includes the list numbers for the possible realizations of the i:th decision variable. The constraints defined by the functions $g_j,\ j \in \mathcal{J}$, are used to remove configurations that are not feasible. Suppose that, for example, $\mathbf{x}' \in X$ should be removed; then, e.g.,

$$g(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} = \mathbf{x}', \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

could be used. Observe that we only assume that $\hat{f}$ is defined for feasible points. As is common for pattern search methods for discrete problems, infeasible points are dealt with by letting $\hat{f}(\mathbf{x}) := M$, where M equals $+\infty$ as in [3], or a large number as in [4]. In other words, we replace the objective function $\hat{f} : \{\mathbf{x} \in X \mid g_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\} \to \mathbb{R}$ with $f : X \to \mathbb{R}$. For general set constraints, for which the only output is whether or not a given point is feasible, such a constraint handling approach is reasonable ([3]).

In practice, we let $M := f(\mathbf{x}^0)$, where $\mathbf{x}^0$ is the first feasible point $\mathbf{x} \in \{\mathbf{x} \in X \mid g_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\}$ that is evaluated. Since the solution method that will be described in Section 4 requires descent in each iteration, this value of M implies that infeasible points are never considered to be local minima for the converted problem (4), which now is defined over a box in $\mathbb{Z}^n$:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}), \\ &\text{subject to} && \mathbf{x} \in X \subseteq \mathbb{Z}^n. \end{aligned} \tag{4}$$

Loosely speaking, ordinary optimization problems (that is, numerical optimization problems) have a decision space which is equipped with a metric, and it is normally at least implicitly assumed that points that are "close" in the decision space are close also in the objective space. This is not true for categorical problems since there is no such metric in general. Since methods for numerical optimization rely on this fact, such methods are not directly applicable to categorical problems.

It must be emphasized that when the variables are transformed into integers when numerizing the problem, one must not over-interpret the numbers. A different translation (ordering) between the original decision variables and the integers results in different problems of the form (4) when viewing them as general nonlinear integer problems. Viewing them as categorical problems they are, however, still equivalent. We have now transformed our pure categorical optimization problem to a class of equivalent general nonlinear integer programs.

General nonlinear integer programming problems belong to the class of NP-hard problems, wherefore all exact algorithms for their solution have exponential computational complexity (unless P = NP). For this reason, it is important to develop approximate methods for the general problem. For classical methods for the general nonlinear integer programming problem, see the survey [11]. One of the interesting members within the class of approximate methods is the recently developed discrete global descent, or filled function, method ([35, 36, 38, 44, 45, 47]), which, as will be shown, can be extended such that a pure categorical problem is not only considered as a general nonlinear integer program, but where the whole class of nonlinear integer programming equivalents of the categorical problem can be exploited. One of the main purposes of this article is to investigate the application of discrete global descent methods for pure categorical optimization problems.

2.1 Why relaxation methods cannot be used for general categorical optimization problems

In [1] an optimization problem containing categorical variables is reformulated into a mixed-integer nonlinear program (MINLP), in which the categorical variables have been removed, and which then can be solved with a standard MINLP solver. To achieve this, the authors, without saying so explicitly, evaluate original fractional points as convex combinations of non-fractional ones. The following example shows why this procedure is not reasonable in general; in particular it is never possible to use when the objective function is a black box.

EXAMPLE 2.1 *Suppose* $n = 2$, $\Omega = \{a, b\} \times \{\alpha, \beta, \gamma\}$ *and consider the optimization problem to*

$$\begin{array}{ll} \text{minimize} & \tilde{f}(\tilde{\mathbf{x}}), \\ \text{subject to} & \tilde{\mathbf{x}} \in \Omega. \end{array}$$

Suppose that $\tilde{f} : \Omega \to \mathbb{R}$ *cannot be evaluated for any* $\tilde{\mathbf{x}} \notin \Omega$ *and that the image of* Ω *is*

$$\tilde{f}(\{(a,\alpha),(a,\beta),(a,\gamma),(b,\alpha),(b,\beta),(b,\gamma)\}) = \{f^1, f^2, f^3, f^4, f^5, f^6\}.$$

Let us now construct a binary reformulation of the problem, with the aim of using some general 0/1-nonlinear optimization algorithm to solve the problem. Let

$$y_{11} = \begin{cases} 1, & \text{if } \tilde{x}_1 = a, \\ 0, & \text{otherwise}, \end{cases} \qquad y_{21} = \begin{cases} 1, & \text{if } \tilde{x}_2 = \alpha, \\ 0, & \text{otherwise}, \end{cases}$$

$$y_{12} = \begin{cases} 1, & \text{if } \tilde{x}_1 = b, \\ 0, & \text{otherwise}, \end{cases} \qquad y_{22} = \begin{cases} 1, & \text{if } \tilde{x}_2 = \beta, \\ 0, & \text{otherwise}, \end{cases}$$

$$y_{23} = \begin{cases} 1, & \text{if } \tilde{x}_2 = \gamma, \\ 0, & \text{otherwise}. \end{cases}$$

The reformulated problem is then to

$$\begin{array}{lll} \textit{minimize} & f(\mathbf{y}), & \\ \textit{subject to} & y_{11} + y_{12} & = 1, \\ & y_{21} + y_{22} + y_{32} & = 1, \\ & y_{i\ell} & \in \{0,1\}, \ i = 1,2, \ \ell = 1,2,3. \end{array}$$

Assume that some method based on continuous relaxation is applied to the reformulated problem. Since $\tilde{f}$ is only computable for $\tilde{\mathbf{x}} \in \Omega$, f cannot be evaluated at fractional points. Using a method of convex combinations as in [1], we have that, e.g., the point $\mathbf{y} = \left[\begin{smallmatrix} y_{11} & y_{12} & \\ y_{21} & y_{22} & y_{23} \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0.3 & 0.7 & \\ 1 & 0 & 0 \end{smallmatrix}\right]$, feasible in the linearly relaxed problem, is evaluated as $f(\mathbf{y}) = 0.3f^1 + 0.7f^4$. In the same way, we have that the point $\mathbf{y} = \left[\begin{smallmatrix} 0.3 & 0.7 & \\ 0.1 & 0.4 & 0.5 \end{smallmatrix}\right]$ is evaluated as $f(\mathbf{y}) = 0.3 \cdot 0.1f^1 + 0.3 \cdot 0.4f^2 + 0.3 \cdot 0.5f^3 + 0.7 \cdot 0.1f^4 + 0.7 \cdot 0.4f^5 + 0.7 \cdot 0.5f^6$, i.e., it is the convex combination of all feasible points.

To summarize, to be able to evaluate *one* point consisting of solely fractional values, one has to compute the value of $\tilde{f}$ for *all* feasible decision vectors, i.e., the whole image $\tilde{f}(\Omega)$ is needed to compute only the value of $f(\mathbf{y})$. This is equivalent to solving the problem by using complete enumeration. In general, with $z_{i\ell} = 1$ if $y_{i\ell}$ is fractional and 0 otherwise, one has to evaluate $\prod_{i=1}^{n} \sum_{i=1}^{m_i} z_{i\ell}$ integral points in order to compute $f(\mathbf{y})$. Obviously, this is not a reasonable method.

3 Optimality conditions and mathematical properties of functions in categorical variables

In this section, we state a number of mathematical properties for pure categorical optimization problems. We start by defining optimality in a categorical setting which then is related to the numerically discrete one. We then move on to show the equivalence of the categorical problem to a whole family of nonlinear integer programming problems. In the Subsection 3.2 we establish mathematical properties of functions defined over categorical spaces that mimic continuity properties of numerical functions; the latter are not available over categorical spaces since there is then no metric in the domain. In the Subsection 3.3 we introduce a property of categorical functions resembling the property of convexity for numerical problems in the sense that local optimality implies global optimality.

3.1 Local and global optimality properties

We begin by defining a locally optimal point in a categorical optimization problem, which is analogous to the classic local optimality definition.

DEFINITION 3.1 (local categorical optimality) *Given a function $f : X \to \mathbb{R}$ and a definition of a (categorical) neighborhood $\mathcal{N}(\mathbf{x}) \subset X$ to a point $\mathbf{x} \in X$, a* local categorical minimum *$\mathbf{x}^* \in X$ is a point such that $f(\mathbf{x}^*) \leq f(\mathbf{y}),\ \mathbf{y} \in \mathcal{N}(\mathbf{x}^*)$.*

In general, $\mathcal{N}(\mathbf{x})$ could be any set-valued function $\mathcal{N} : X \to 2^X$, where for some applications including categorical variables certain choices could be natural. In this article, we use the Hamming metric

$$d_{\text{H}}(\mathbf{x}, \mathbf{y}) = \text{card}\left\{i \in \{1, \ldots, n\} \mid x_i \neq y_i\right\},$$

to define the *categorical neighborhood* to a point $\mathbf{x} \in X$ as

$$\mathcal{N}(\mathbf{x}) = \{\mathbf{y} \in X \mid d_{\text{H}}(\mathbf{x}, \mathbf{y}) \le 1\}, \tag{5}$$

which seems reasonable when considering a problem as the one in Example 1.1 when no special structure of the objective function over the domain is known. The *discrete neighborhood* (where there is an order of the variables in the decision space) is naturally defined as

$$\mathcal{N}_d(\mathbf{x}) = \big\{\{\mathbf{x}\} \cup \mathbf{y} \in X \mid \mathbf{y} = \mathbf{x} \pm \mathbf{d},\ \mathbf{d} = \mathbf{e}_i,\ i = 1, \dots, n\big\}. \tag{6}$$

Using the discrete neighborhood (6) instead of the categorical neighborhood (5), Definition 3.1 defines an ordinary *local discrete minimum*.

A connectedness property for integer sets is defined according to the following.

DEFINITION 3.2 (pathwise connected set) *A set $X \subseteq \mathbb{Z}^n$ is called a* pathwise connected set *if for all pairs of distinct points $\mathbf{x}'$ and $\mathbf{x}''$ in X, there is a sequence of steps $\{\mathbf{x}^i\}_{i=0}^{u} \subseteq X$ such that $\mathbf{x}' = \mathbf{x}^0$, $\mathbf{x}'' = \mathbf{x}^u$ and $\|\mathbf{x}^{i+1} - \mathbf{x}^i\| = 1,\ i = 1, \dots, u-1$.*

Two obvious implications are collected in the following proposition.

PROPOSITION 3.3 (categorical minimum implies discrete minimum) *Suppose $\mathbf{x}^*$ is a (strict) local categorical minimum of f on X with respect to the neighborhood definition* (5)*. Then $\mathbf{x}^*$ is also a (strict) local discrete minimum of f on X. If $\mathbf{x}^*$ is a strict local discrete minimum of f on X, and all level sets of the image $f(X)$ are pathwise connected, then $\mathbf{x}^*$ is also a strict categorical local minimum of f on X.*

DEFINITION 3.4 (sorting) *Let π_i be a permutation of $\{1, \dots, m_i\}$ and collect the permutations in $\Pi = (\pi_1, \dots, \pi_n)$. A* sorting *of a variable $\mathbf{x} \in X$ is then defined as $\Pi\mathbf{x} := \big(\pi_1(x_1), \dots, \pi_n(x_n)\big)$.*

PROPOSITION 3.5 (categorical equivalence) *Consider the problems to*

$$(P) \quad \begin{array}{ll} \textit{minimize} & f(\mathbf{x}), \\ \textit{subject to} & \mathbf{x} \in X \subseteq \mathbb{Z}^n, \end{array} \qquad \textit{and} \qquad (P_\Pi) \quad \begin{array}{ll} \textit{minimize} & f_\Pi(\mathbf{x}), \\ \textit{subject to} & \mathbf{x} \in X \subseteq \mathbb{Z}^n. \end{array} \tag{7}$$

for some collection of permutations Π, and where we use the notation $f_\Pi(\mathbf{x}) := f(\Pi x)$. Given the definition (5) *of a categorical neighborhood, the problems (P) and (P_Π) are categorically equivalent in the sense that corresponding points in the respective problem have the same categorical neighborhoods. Furthermore, the numbers of local categorical minima in the two problems coincide.*

Proof The result is a consequence of the identity $\mathcal{N}(\mathbf{x}) = \mathcal{N}(\Pi\mathbf{x}),\ \mathbf{x} \in X$, which holds since $d_{\text{H}}(\mathbf{x}, \mathbf{y}) = d_{\text{H}}(\Pi\mathbf{x}, \Pi\mathbf{y}),\ \mathbf{x}, \mathbf{y} \in X$. ∎

As a remark, we note that the number of local discrete minima in categorically equivalent problem formulations might be very different, as illustrated in the following example.

EXAMPLE 3.6 *Consider the problem (P) defined by the objective space to the left in Figure 1 below. This problem has 8 local discrete minima (marked with a bold font), where 2 of these (the ones with $f(\mathbf{x}) = 1$) are categorical local minima. If the variables are sorted according to $\Pi = (\{1,3,2,4\}, \{1,3,2,4\})$, then the objective space to the sorted problem (P_Π) becomes as to the right.*

$$f(X) = \begin{bmatrix} 4 & \mathbf{1} & 5 & \mathbf{2} \\ \mathbf{2} & 3 & \mathbf{3} & 4 \\ 3 & \mathbf{2} & 4 & \mathbf{3} \\ \mathbf{1} & 4 & \mathbf{2} & 5 \end{bmatrix} \qquad f_\Pi(X) = \begin{bmatrix} 4 & 5 & \mathbf{1} & 2 \\ 3 & 4 & 2 & 3 \\ 2 & 3 & 3 & 4 \\ \mathbf{1} & 2 & 4 & 5 \end{bmatrix}$$

Figure 1: An example showing the invariance of the number of categorical minima, but not of the discrete minima, with respect to sorting.

We see that (P_Π) still has 2 local categorical minima, but these are also the only local discrete minima to this problem.

We will return to permutations and suggestions on how to construct desirable permutations in Section 4.3.2.

In optimization of numerical functions, one implicitly assumes a regularity property of the objective function in order to be able to infer information from evaluated points in the domain to non-evaluated ones; example properties are linearity, continuity, continuity almost everywhere, Lipschitz continuity, and convexity. To be able to draw inferences also in categorical problems, and make other methods than complete enumeration meaningful, regularity properties of a similar kind are necessary. The aim of the following subsections is to suggest a number of such properties.

3.2 Continuity-like properties

The standard continuity definition for functions between metric spaces would, for our metric spaces (X, d_H) and $(\mathbb{R}, |\cdot|)$, require that for f to be continuous in X, then for all $\mathbf{a} \in X$, and for each $\varepsilon > 0$ there must be a $\delta > 0$ such that $d_H(\mathbf{x}, \mathbf{a}) < \delta$ implies $|f(\mathbf{x}) - f(\mathbf{a})| < \varepsilon$. This property holds for any function f for any $\delta \in (0, 1)$, i.e., all categorical functions are continuous with respect to the neighborhood definition given by the Hamming metric. Therefore, the standard continuity definition is not an appropriate tool for deciding if a function over a categorical space is well-behaved or not.

The intuitive interpretation of continuity for $f : X \to \mathbb{R}$ is that images of points that are close in X are close in $\mathbb{R}$. The smallest reasonable neighborhood of a categorical vector $\mathbf{x}$ containing more points than $\mathbf{x}$ itself (by only using information from the decision space X) is the suggested neighborhood $\mathcal{N}(\mathbf{x})$ defined in equation (5). Therefore, a natural requirement of a continuity-like definition for functions over a categorical space is that a pair of points belonging to the same neighborhood are more likely to be close in the objective space than an arbitrary pair.

However, it is natural to argue that it does not matter if there are some points in the neighborhood $\mathcal{N}(\mathbf{x})$ that are far away from $\mathbf{x}$ in the objective space, if there are others that are close. If the problem has a structure which would be revealed with some sorting, then the definition (5) of $\mathcal{N}(\mathbf{x})$ is natural, since the sorting is unknown; however, for a continuity-like property of f it would suffice to have some point in the neighborhood close to $\mathbf{x}$. A formal definition of such a property is given below.

DEFINITION 3.7 (property *P1*) *Let $\mathbf{x}, \mathbf{z}^1, \ldots, \mathbf{z}^{card\,\mathcal{N}(\mathbf{x})} \in X$ be selected from uniform distributions and let $\mathbf{y}' \in \arg\min_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} |f(\mathbf{x}) - f(\mathbf{y})|$. Then, if*

$$\mathbb{E}\left[|f(\mathbf{x}) - f(\mathbf{y}')|\right] < \mathbb{E}\Big[\min_{\ell=1,\ldots,card\,\mathcal{N}(\mathbf{x})} |f(\mathbf{x}) - f(\mathbf{z}^\ell)|\Big]$$

holds, we say that the property P1 *holds for f over X.*

Since the ordinary continuity property implies closeness in all directions, we could require that the above property should hold not just for any point $\mathbf{y}$ in the neighborhood of $\mathbf{x}$, but if there is a structure in all dimensions, then the closeness should also hold in all dimensions of the decision space. Such a property would, however, be unnecessarily strong. The continuity-like property should be related to whether a neighborhood search method, as presented as a local search method for categorical problems in Section 4.1, is reasonable. Property *P1* should be enough for such a method to be applicable. In continuous optimization there is normally an underlying assumption of a limited multi-modality of the objective functions considered. Then, by evaluating the points in a neighborhood around a current iterate, the likelihood of finding an improving point is larger than if evaluating the same number of points, but randomly distributed, in the domain. In the categorical setting, the above underlying assumption could be interpreted as that *P1* holds.

We suggest also a second property that instead of resembling the standard continuity definition focuses more on the structure of f over X. The idea is that a sorting that is done with local information of f in a neighborhood can be inferred to the whole domain. This property should hold if the second extension of the discrete global descent method presented in Section 4.3.2 should be meaningful. Formally, this property is defined according to the following.

DEFINITION 3.8 (property *P2*) *For each $i \in \{1, \ldots, n\}$, let $\mathbf{x} \in X$, $\ell \in \{1, \ldots, n\} \setminus \{i\}$ and $y_\ell \in \{1, \ldots, m_\ell\}$ be selected from uniform distributions. Consider $\mathcal{N}_i = \mathcal{N}_i(\mathbf{x})$, which is a line in X. Create a sorting of the variables by constructing a collection of permutations Π at $\mathbf{x}$ such that f increases along all dimensions of $\mathcal{N}(\Pi\mathbf{x})$. Let $\mathcal{N}_i' := \{\mathbf{y} \in X \mid \mathbf{y} = \mathbf{z} + \mathbf{e}_\ell,\, \mathbf{z} \in \mathcal{N}_i(\Pi\mathbf{x})\}$, where $\mathbf{e}_\ell$ denotes the unit vector with the value 1 in the ℓ:th position where $e_{\ell_\ell} \in \{-1, 1\}$ is selected such that $|f(\Pi\mathbf{x}) - f(\Pi\mathbf{x} + \mathbf{e}_\ell)|$ is minimal. This is the line along the i:th dimension of X and shifted in the ℓ:th dimension, which is closest to $\mathcal{N}_i(\mathbf{x})$ with* *respect to the information gained by computing f over $\mathcal{N}(\mathbf{x})$. An arbitrary line is likewise defined as $\mathcal{N}_i'' = \{\mathbf{y} \in X \mid y_j = z_j, \forall j \neq \ell, \mathbf{z} \in \mathcal{N}_i(\mathbf{x})\}$ (see Figure 2 for an illustration). Then, if*

$$\mathbb{E}\big[\|f(\mathcal{N}_i(\mathbf{x})) - f(\mathcal{N}_i(\mathbf{x})'\|\big] < \mathbb{E}\big[\|f(\mathcal{N}_i) - f(\mathcal{N}_i'')\|\big], \quad i = 1, \ldots, n,$$

where the norm is Euclidean, but where the element corresponding to x_i is not considered, holds, we say that the property P2 *holds for f over X.*

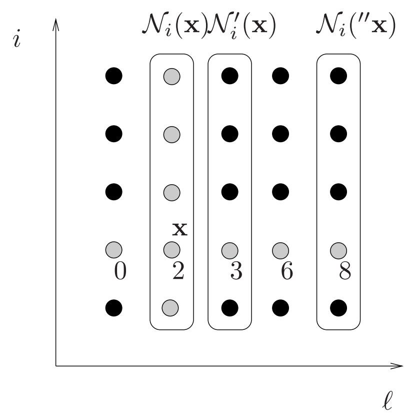


Figure 2: Illustration of the Property *P2*. The dots represent the points in the sorted domain, and the gray dots represent $\mathcal{N}(\mathbf{x})$. The numbers are examples of function values that would lead to this sorting. If property *P2* holds, then the neighboring lines $\mathcal{N}_i(\mathbf{x})$ and $\mathcal{N}_i'(\mathbf{x})$ in the sorted domain would be more similar than an arbitrary pair of lines (such as $\mathcal{N}_i(\mathbf{x})$ and $\mathcal{N}_i''(\mathbf{x})$).

To summarize this subsection, we have suggested mathematical properties for functions defined over categorical domains. The property *P1* is related to if the neighborhood definition is reasonable for the function, and thereby if a neighborhood search method is meaningful for the optimization of a categorical problem. Property *P2* is related to if the behavior of the function locally within a neighborhood is similar to the behavior in neighborhoods elsewhere in the domain. Such a property is necessary for the solution method to be presented in Section 4.3.2. It is reasonable to believe that many real-world problem possess these properties, and we assume that they hold for the categorical problems that we want to solve in what follows.

3.3 A convexity-like property

An important concept in continuous optimization is convexity and a central result for continuous optimization problems is that if the objective function is convex over a convex feasible set, then a local minimum is also a global minimum.

The concept of convexity has also in various ways been generalized to (numerically) discrete domains. Some well-known concepts are discretely-convexity [29], integrally convexity [13], $M^\natural$-convexity and $L^\natural$-convexity [31, 32]. For definitions and properties of the concepts, we refer to the references. These generalizations all carry the important property that some local minimality implies global minimality.

In [34], relations between the above types of convexity concepts are analyzed. It is shown that the strict inclusions

$$\text{Discretely-convexity} \supset \text{Integrally convexity} \supset M^\natural\text{- and } L^\natural\text{-convexity} \tag{8}$$

hold. For all these concepts, the max-norm is somehow used to define local neighborhoods or allowed perturbations around a point such that "diagonal" steps can be taken. Using the max-norm in the definition of discrete neighborhoods is, however, not natural in our application, since such a neighborhood will not be a subset of the categorical neighborhood. We suggest a new type of discrete convexity property suited to categorical domains and for which a similar important relation between local and global minima holds.

DEFINITION 3.9 (categorical convexity) *Given a function $f : X \to \mathbb{R}$, if there exists a collection of permutations Π such that all level sets to $f_\Pi(X)$ are pathwise connected, then f is* categorically convex *over X.*

PROPOSITION 3.10 (convexity implication) *Suppose that f is a categorically convex function over its domain X. If $\mathbf{x}^* \in X$ is a* strict *local categorical minimum to (P_Π), then $\mathbf{x}^* = \mathbf{x}^*_{\text{glob}}$ is a unique global minimum to (P) (and (P_Π)).*

Proof Suppose that $\mathbf{x}^* \in X$ is a strict local categorical minimum to (P_Π) and that all level sets to $f_\Pi(X)$ are pathwise connected. Then all points in $\mathcal{N}(\mathbf{x}^*) \setminus \{\mathbf{x}^*\}$ have strictly larger function values than $\mathbf{x}^*$, wherefore $\mathbf{x}^*$ is an isolated point in the level set $\{\mathbf{x} \in \mathcal{N}(\mathbf{x}^*) \mid f(\mathbf{x}) \leq f(\mathbf{x}^*)\}$.

Suppose now that $\mathbf{x}^*$ is not a unique global minimum. Then $\exists \mathbf{y} \in X \setminus \mathcal{N}(\mathbf{x}^*)$ such that $f_\Pi(\mathbf{y}) \leq f_\Pi(\mathbf{x}^*)$. But there is no discrete path connecting $\mathbf{x}^*$ and $\mathbf{y}$ in the level set defined by $\{\mathbf{x} \in X \mid f_\Pi(\mathbf{x}) \leq f(\mathbf{x}^*)\}$. This contradicts the assumption that the level sets of $f_\Pi(X)$ are pathwise connected, and therefore the result is proved by contradiction. ∎

REMARK 3.11 *Strictness is necessary in the assumptions in Prop. 3.10. This is shown with the example $f(X) = \left[\begin{smallmatrix} 2 & 2 \\ 2 & 1 \end{smallmatrix}\right]$, where the original sorting leads to pathwise connected level sets and where the upper left element is a (non-strict) local categorical minimum but not a global minimum.*

REMARK 3.12 *The converse result of Proposition 3.10 is not true, i.e., that each strict local categorical minimum is a global minimum does not imply that there is a sorting such that all level sets of $f_\Pi(X)$ are pathwise connected. A counterexample is given by the objective space $f(X) = \left[\begin{smallmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{smallmatrix}\right]$, in which the bottom left element is the single strict local categorical minimum and also the unique global minimum, but where none of the possible collections of permutations lead to connected level sets.*

There are no inclusion relationships between any of the previously defined convexity concepts for discrete domains presented above and the categorical convexity concept, as shown by the following proposition. This motivates the introduction of the new convexity-like property.

PROPOSITION 3.13 (inclusion relationships) *There are no inclusion relationships between any of the concepts of discretely-convexity, integrally convexity, $M^\natural$-convexity, and $L^\natural$-convex with the concept of categorical convexity.*

Proof The proof is constructed by the use of exemplifying functions. The function f with the image $f(X) = \left[\begin{smallmatrix} 0 & 1 & 2 \\ 2 & 1 & 1 \end{smallmatrix}\right]$ is categorically convex but not discretely-convex, as easily shown from the definition in [29]. From (8) we conclude that the class of categorically convex functions is not included in the class of integrally convex, neither among $M^\natural$-convex or $L^\natural$-convex, functions.

The function f with the image $f(X) = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ is $L^\natural$-convex since it possesses the mid-point convexity property ([34]) which in [16] is shown to be equivalent to $L^\natural$-convexity. From the relations (8) it is clear that f is also integrally convex and discretely-convex. However, it is not categorically convex.

Finally, the function f with the image $f(X) = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right]$ is $M^\natural$-convex since it fulfills its so called exchange properties ([33]). However, it is not categorically convex. ∎

To conclude this subsection, we note that we have in Definition 3.9 suggested a new convexity-like property, categorical convexity, for functions defined on discrete domains. This property is suited for categorical problems, and has the implication that a strict local categorical minimum is also a unique global minimum. We have also shown that the new property has no inclusion properties with some previous convexity concepts for functions defined over discrete domains.

It is at least as unlikely that a real-world categorical problem is categorically convex as that a real-world continuous problem is convex, unless the former problem has some special structure. And even if it has the property, no efficient method is known for checking whether it holds or not. Still, the result is important in itself for the foundations of optimality in pure categorical optimization problems, and it is possible that the concept can be exploited in future solution methods.

As opposed to the continuity-like properties presented in the previous subsection, we do not assume that the pure categorical problems have the convexity-like property *P2* for the suggested solution methods.

4 Solution methods

The purpose of this article is to show that pure categorical optimization problems can be solved using mathematical techniques. In this section, we suggest a particular method.

4.1 Finding local minimizers to categorical optimization problem

Recall Definition 3.1 of a local minimizer to a categorical problem. Since there is no distance measure in the domain for a categorical problem, there are no "directions". Therefore, all types of line search or gradient based methods for the solution of a categorical problem are disqualified. It is instead natural to employ a direct (or pattern) search technique, in which one iteratively computes the objective function in the categorical neighborhood (5) of the current point, moving to an improved point if one is found. This process is then repeated until the current point is the best one in its categorical neighborhood, and hence is a local minimizer according to Definition 3.1. The steps of this simple descent algorithm are shown in Algorithm 1 below.

Algorithm 1 Single-objective algorithm for local (categorical) minimizers

1: Pick starting point $\mathbf{x} = \mathbf{x}^0$ and let $\bar{f} = f(\mathbf{x})$.
2: compute $\mathcal{N}(\mathbf{x})$.
3: **for** $\mathbf{x}^i \in \mathcal{N}(\mathbf{x}) \setminus \{\mathbf{x}\}$ **do**
4:     **if** $f(\mathbf{x}^i) < \bar{f}$ **then**
5:         Let $\bar{f} = f(\mathbf{x}^i)$ and return to 2 with $\mathbf{x} \leftarrow \mathbf{x}^i$.
6:     **end if**
7: **end for**
8: Return the locally optimal point $\mathbf{x}^* = \mathbf{x}$ with objective value $\bar{f}$.

Since the method requires strict descent in each iteration, and since the domain is finite, the method converges in a finite number of steps. The method can in the worst-case, however, require that all points in the domain are computed, even for a well-behaved categorical problem fulfilling the categorical convexity property of Definition 3.9. An example is given by the objective space

$$f(X) = \begin{bmatrix} \mathbf{5} & 5 & 5 & \mathbf{4} \\ 6 & \mathbf{1} & \mathbf{0} & 4 \\ 6 & \mathbf{2} & 3 & \mathbf{3} \\ \mathbf{6} & 6 & 6 & 6 \end{bmatrix}, \tag{9}$$

where the lower left point (with $f(\mathbf{x}) = 6$) is selected as the starting point. The bold-face numbers correspond to the improving points in the six iterations needed for convergence (to a vector $\mathbf{x}$ with $f(\mathbf{x}) = 0$). The union of the categorical neighborhoods of the sets of bold-face iteration points equals the whole domain, wherefore all points in the domain have to be computed before finding an optimal solution.

This worst-case behavior seems unavoidable; however, it is not unique for categorical problems: if we instead consider a numerical discrete problem with the discrete neighborhood definition given by (6) and a problem instance given by the objective space

$$f(X) = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ 4 & 4 & 4 & \mathbf{4} \\ 9 & 8 & 7 & \mathbf{5} \\ \mathbf{9} & \mathbf{8} & \mathbf{7} & \mathbf{6} \end{bmatrix}, \tag{10}$$

then the same negative result holds in the sense that there are functions requiring that the whole domain has to be computed until a local minimum is found.

If the categorical problem possesses the property of categorical convexity (Definition 3.9), then we know from Proposition 3.10 that if the local minimum obtained is strict, it is also the unique global minimum to the problem. However, for a general pure categorical problem, categorical convexity is not expected to hold, wherefore we need a procedure not only for finding a local minimum, but to find a *good* local minimum. The following section presents a method for finding (hopefully) good local minima to general nonlinear integer programs, and in the section thereafter we adapt this method to problems of the categorical type.

4.2 Discrete global descent methods

Global descent methods (or filled function methods) were introduced in the 1980s as a way of solving nonlinear continuous optimization problems ([17]). The main idea is to repeatedly descend from one local minimum to another and better one, until a global optimum is reached. Once a local minimum has been determined by some local optimization method, an auxiliary function is constructed such that for this auxiliary function the current local minimizer is a local maximizer (we say that the basin of attraction of the local minimum is "filled"), and it should have no minimizers in any basin of f which is higher than the current local minimum. By minimizing the auxiliary function starting from this local maximizer, the hope is to find a point in a basin of attraction of a minimum to f lower than the previous one.

In, e.g., [18], the global descent method is applied to discrete problems by converting the discrete problem to an approximate continuous counterpart before applying the (continuous) global descent method. More recently, several types of global descent methods have been developed for nonlinear (numerical) discrete optimization problems ([35, 36, 38, 44, 45, 47]), handling the discreteness explicitly, with varying constructions of the auxiliary function and with varying theoretical properties. In the article [43], a review of various suggested discrete filled function methods in terms of theoretical properties and computational efficiency is provided. Applicability, reliability and efficiency of this global optimization technique is verified, and from the computational experiments it is concluded that, from the variants that are implemented, the algorithm from [35] works the best. We summarize this version of the method next.

In [35] the problem

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}), \\ &\text{subject to} \quad \mathbf{x} \in X, \end{aligned} \tag{11}$$

is considered and X is assumed to be a finite and pathwise connected set (see Definition 3.2). The authors define a *discrete global descent function* at a local minimizer $\mathbf{x}^*$ of f to be a function $G_{\mathbf{x}^*} : X \to \mathbb{R}$ satisfying the conditions

(*1.*) $\mathbf{x}^*$ is a strict local maximizer of $G_{\mathbf{x}^*}$ over X,
(*2.*) $G_{\mathbf{x}^*}$ has no local minimizers in $\{\mathbf{x} \in X \mid f(\mathbf{x}) \geq f(\mathbf{x}^*)\} \setminus X^c$, and

(3.) $\mathbf{x}^{**} \in X \setminus X^c$ is a local minimizer of f over X with $f(\mathbf{x}^{**}) < f(\mathbf{x}^*)$ if and only if $\mathbf{x}^{**}$ is a local minimizer of $G_{\mathbf{x}^*}$ over X,

where X^c is the set of corner points of X, i.e.,

$$X^c := \big\{\mathbf{x} \in X \mid \mathbf{x} + \mathbf{d} \in X \Rightarrow \mathbf{x} - \mathbf{d} \notin X,\ \mathbf{d} \in \{\pm e_i \mid i = 1, \dots, n\}\big\}.$$

A family of two-parameter discrete global descent (or filled) functions are proposed as the auxiliary function constructed at a local minimum $\mathbf{x}^*$ of (11) as

$$G_{\mu,\rho,\mathbf{x}^*}(\mathbf{x}) = A_\mu\left(f(\mathbf{x}) - f(\mathbf{x}^*)\right) - \rho\|\mathbf{x} - \mathbf{x}^*\|, \text{ in which} \tag{12}$$
$$A_\mu(y) = y\mu\Big[(1-c)\Big(\frac{1-c\mu}{\mu - c\mu}\Big)^{-y/\tau} + c\Big],$$

where $c \in (0,1]$ and $\tau > 0$ is a sufficiently small number, theoretically required to be smaller than $\min\{|f(\mathbf{x}') - f(\mathbf{x}'')| : \mathbf{x}', \mathbf{x}'' \in X,\ f(\mathbf{x}') \neq f(\mathbf{x}'')\}$. When certain conditions of the parameters μ and ρ are satisfied, then the global descent function has the desired properties (1.)–(3.) above; these conditions, however, require that μ and ρ are sufficiently small compared to certain values defined by, among others, the problem dependent Lipschitz constant

$$L := \max_{\mathbf{x}',\mathbf{x}'' \in X} \frac{|f(\mathbf{x}') - f(\mathbf{x}'')|}{\|\mathbf{x}' - \mathbf{x}''\|},$$

which in general is impossible to know a priori. Furthermore, even if the conditions on μ and ρ are satisfied, it is not certain that one is guaranteed to succeed in finding a minimum $\mathbf{x}^{**}$ of $G_{\mu,\rho,\mathbf{x}^*}$ with $f(\mathbf{x}^{**}) < f(\mathbf{x}^*)$), even if such a minimum exists, since one can end up in X^c, i.e., in corner points of X, when minimizing $G_{\mu,\rho,\mathbf{x}^*}$ starting from $\mathbf{x}^*$.

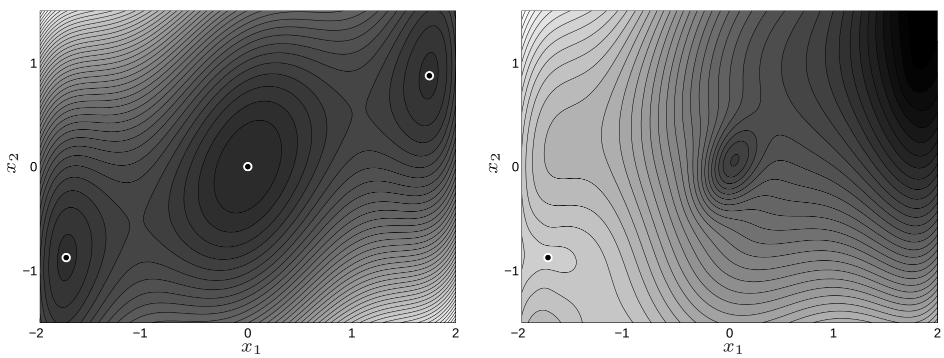
In the algorithm, both parameters μ and ρ are initialized to some small values. These are reduced sequentially if no improving points for f are obtained when minimizing $G_{\mu,\rho,\mathbf{x}^*}$. If the parameter values are decreased below preset threshold values, then the last local minimum $\mathbf{x}^* = \mathbf{x}^*_{\text{final}}$ is taken as an approximation of the global minimum $\mathbf{x}^*_{\text{glob}}$. As shown by a number of numerical examples, the resulting approximate global minimum is often a true global minimum. In Algorithm 2, the principal steps of the discrete global descent method implemented in [35] are shown. We refer to the original article for the full description. In Figure 3, the discrete global descent function is illustrated using the values $\mu = \rho = 0.01$ for the integrality-constrained modification (13) of the 3-hump camel back problem proposed in [12] in its original form, and used for illustrative purposes also in, e.g., [35, 43]. This problem has one global minimum at $\mathbf{x}^* = (0,0)^{\mathrm{T}}$ and two non-global local minima at $\mathbf{x}^* = \pm(1.748, 0.874)$.

$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) := 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2,\\
\text{subject to} \quad & x_i = \frac{y_i}{1000}, \quad i = 1, 2,\\
& -2000 \le y_1 \le 2000,\\
& -1500 \le y_2 \le 1500,\\
& y_1, y_2 \in \mathbb{Z}.
\end{aligned} \tag{13}$$

Algorithm 2 Discrete global descent method

1: Initialize the parameters c, τ, μ and ρ, and decide on reduction strategies and thresholds $\hat{\mu}$, $\hat{\rho}$ for the latter two. Pick a starting point $\mathbf{x} = \mathbf{x}^0 \in X$.
2: Find a local minimizer $\mathbf{x}^*$ of f starting from $\mathbf{x}$.
3: Compute $G_{\mu,\rho,\mathbf{x}^*}$.
4: **for** $\mathbf{x}^i \in \mathcal{N}_d(\mathbf{x}^*) \setminus \{x^*\}$ **do**
5:     Start minimizing $G_{\mu,\rho,\mathbf{x}^*}$ starting from $\mathbf{x}^i$ leading to an iteration sequence $\{\mathbf{x}^{i,k}\}_{k=1}^{N}$.
6:     **if** for some k, $f(\mathbf{x}^{i,k}) < f(\mathbf{x}^*)$ **then**
7:         Return to 2 with $\mathbf{x} \leftarrow \mathbf{x}^{i,k}$.
8:     **end if**
9: **end for**
10: **if** $\mu > \hat{\mu}$ or $\rho > \hat{\rho}$ **then**
11:     Reduce parameters according to the reduction strategy. Return to 3.
12: **end if**
13: Take $\mathbf{x}^* = \mathbf{x}^*_{\text{final}}$ as an approximation of a global optimum.

As noted in [43], the typical way of handling constraints in discrete global descent



(a) Level curves for $f(\mathbf{x})$ in (13). The three local minima are marked with black dots with white borders.

(b) Level curves for $G_{\mu,\rho,\mathbf{x}^*}(\mathbf{x})$ for the local minimum $\mathbf{x}^* = (-1.748, -0.874)^{\mathsf{T}}$. The point $\mathbf{x}^*$ is marked with a black dot with a white border.

Figure 3: Illustration of the global descent function for the 3-hump camel back function.

approaches is by representing them by penalty terms in the objective, thus converting the constrained problem to one with a pathwise connected feasible set; the latter property is needed for many existence results such as that there is a global descent function fulfilling the properties (*1.*)–(*3.*). One exception, however, is the method described in [44], where constraints are included explicitly in the global descent function.

4.3 Global descent methods for categorical problems

To the principal steps of a discrete global descent method we suggest two extensions in order to adapt it to categorical problems. In this article, we have focused on the discrete global descent method as presented in [35]; however, the extensions should be valid for any variant of the method.

4.3.1 Categorical local search

The first necessary extension of the principal method is due to the fact that the problem is not numerical but categorical. In the latter case it is not reasonable to use the discrete neighborhood definition (6) in the local search of f, since the ordering of the variables is arbitrary, wherefore points in $\mathcal{N}_d(\mathbf{x})$ should in general not be any "closer" to $\mathbf{x}$ than a point which lies in $\mathcal{N}(\mathbf{x}) \setminus \mathcal{N}_d(\mathbf{x})$. Therefore, we suggest that the local search, i.e., Step 2 should be performed with respect to the categorical neighborhood, i.e., using Algorithm 1.

4.3.2 Sorting

The second extension of the principal method builds on the fact that different permutations of the variables lead to categorically equivalent, but not numerically equivalent, problems. The idea is to find good sortings such that the numerized problems are as well-behaved as possible in the sense that they are easy to solve using a method for numerical optimization, in particular the discrete global descent method. We assume that the categorical problem has some underlying structure, corresponding to property *P2* (see Definition 3.8), which we, at least partly, hope to be able to reconstruct by computing the function values at a limited number of points in the domain. Specifically, each time we arrive at Step 3 in Algorithm 2, standing in the local minimum $\mathbf{x}^*$, the function values already computed in $\mathcal{N}(\mathbf{x}^*)$ are utilized to construct sortings.

We suggest two variants of sorting procedures, *Increasing* and *Central*. For both variants, a permutation π_i, $i = 1, \ldots, n$, is constructed for each dimension of the domain, and the sorting given by $\Pi = (\pi_1, \ldots, \pi_n)$ is collected.

Let $\mathcal{N}_i(\mathbf{x})$ be the neighborhood of $\mathbf{x} \in X$ in the i:th dimension, with the corresponding image $f(\mathcal{N}_i(\mathbf{x}))$. In the *Increasing* procedure, we let π_i be the ordering of $f(\mathcal{N}_i(\mathbf{x}))$ from smallest to largest, where ties are broken arbitrarily. With this sorting procedure, the number of strict local *discrete* minimum points of f restricted to $\mathcal{N}(\mathbf{x})$ is at most one. For example, if $f(\mathcal{N}_i(\mathbf{x})) = (1, 3, 2, 5, 2)$, then the permutation π_i should transform

$$(1, 2, 3, 4, 5) \xrightarrow{\pi_i} (1, 3, 5, 2, 4) \text{ or } (1, 2, 3, 4, 5) \xrightarrow{\pi_i} (1, 5, 3, 2, 4)$$

such that $f(\pi_i \mathcal{N}_i(\mathbf{x})) = \pi_i f(\mathcal{N}_i(\mathbf{x})) = (1, 2, 2, 3, 5)$.

Since the sorting will always be performed at a local categorical minimum of f, the *Increasing* procedure will permute the domain such that the current point $\mathbf{x} = \mathbf{x}^*$ is at a corner of the sorted domain. In the discrete global descent method, the hope is to find an improved local minimum in the interior of (or at least not in a corner of) X; hence, it might be better to design a sorting procedure that positions $\mathbf{x}^*$, the starting point for the minimization of $G_{\mu,\rho,\mathbf{x}^*}$, at somewhere central in the domain. This is done in the *Central* procedure, where π_i is constructed such that $f(\pi_i \mathcal{N}_i(\mathbf{x}))$ monotonically decreases in the first half of the vector and monotonically increases in the second. There are always multiple sortings fulfilling this property (in contrast to the former procedure) since for each point it is not determined if it should be positioned in the decreasing or in the increasing part of $f(\pi_i \mathcal{N}_i(\mathbf{x}))$. The *Central* procedure is designed such that if $\mathbf{x} = \mathbf{x}^*$ is a local categorical minimum, then it is always positioned as close as possible to the mid point of the sorted domain.

The two procedures *Increasing* and *Central*, respectively, are shown as implemented in Algorithm 3 and Algorithm 4.

Algorithm 3 The *Increasing* sorting procedure at a point $\mathbf{x} \in X$

for $i \in \{1, \ldots, n\}$ **do**
  Retrieve the function values $f(\mathcal{N}_i(\mathbf{x})) := \left(f^1_{\mathcal{N}_i(\mathbf{x})}, \ldots, f^{m_i}_{\mathcal{N}_i(\mathbf{x})}\right)$ whose original order is given by the value of x_i.
  Sort the vector $f(\mathcal{N}_i(\mathbf{x}))$ from the smallest to the largest value. If there are ties, consider the earlier point in the original order as the smallest.
  Let π_i be the ordering of $f^1_{\mathcal{N}_i(\mathbf{x})}, \ldots, f^{m_i}_{\mathcal{N}_i(\mathbf{x})}$ in the sorted vector.
end for
Collect $\Pi = (\pi_i, \ldots, \pi_n)$.

Algorithm 4 The *Central* sorting procedure at a point $\mathbf{x} \in X$

for $i \in \{1, \ldots, n\}$ **do**
  Retrieve the function values $f(\mathcal{N}_i(\mathbf{x})) := \left(f^1_{\mathcal{N}_i(\mathbf{x})}, \ldots, f^{m_i}_{\mathcal{N}_i(\mathbf{x})}\right)$ whose original order is given by the value of x_i.
  Take the smallest element of $f(\mathcal{N}_i(\mathbf{x}))$ and let the sorted vector contain only this element. Then take the next smallest element of $f(\mathcal{N}_i(\mathbf{x}))$ and extend the sorted vector with this element in front of the first. Continue by taking the smallest remaining element of $f(\mathcal{N}_i(\mathbf{x}))$ and put every other smallest element in the front and every other smallest element in the rear of the partially built sorted vector. If there are ties, consider the earlier point in the original order as the smallest.
  Let π_i be the ordering of $f^1_{\mathcal{N}_i(\mathbf{x})}, \ldots, f^{m_i}_{\mathcal{N}_i(\mathbf{x})}$ in the sorted vector.
end for
Collect $\Pi = (\pi_i, \ldots, \pi_n)$.

By using the two sorting procedures, the problems (P_Π) that are constructed have, at least within the neighborhoods of the current point at which the sorting has been performed, a well-behaved structure. If the assumption that there is an underlying structure in the problems such that the function behavior within neighborhoods is similar among neighborhoods holds true, then it is expected that there will be some structure of the sorted function in the rest of the domain as well.

In Figure 4, the two sorting procedures are illustrated on the discretized version (14) of the standard Goldstein–Price test problem for global optimization presented in [19] in its original form:

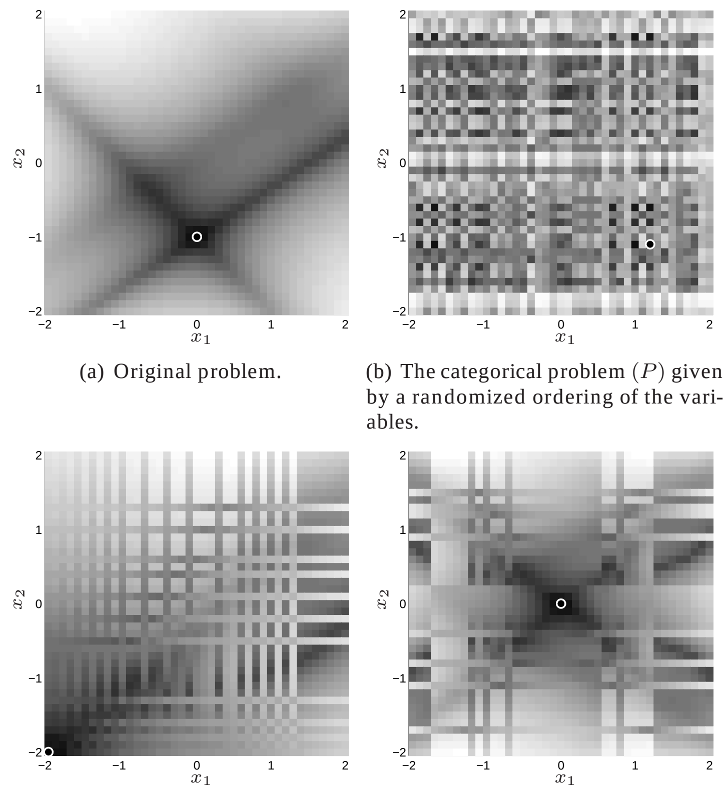
$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) := \log\Big(\big(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\big) \\
& \qquad \cdot\big(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\big)\Big), \\
\text{subject to} \quad & x_i = \frac{y_i}{10}, \qquad i=1,2, \\
& -20 \le y_i \le 20, \qquad i=1,2, \\
& y_1, y_2 \in \mathbb{Z}.
\end{aligned} \tag{14}$$

In Figure 4(a) the original objective space of the problem is shown; it clearly has a structure. In Figure 4(b) a categorical problem (P) has been mimicked with an arbitrary ordering of the variables constructed in each dimension. In the Figures 4(c) and 4(d) the objective spaces of the sorted problems (P_Π) obtained using the *Increasing* and the *Central* sorting procedures, respectively, are shown. In the latter two figures, we see that the structure of the original (unknown) problem can partially be reconstructed. To evaluate the "well-behavior" of the different numerized functions, one option is to compute the number of discrete local minima. For this problem, the original problem has 12 local categorical minima and 42 unique local discrete minima. The categorical problem illustrated in Figure 4(b) still has 12 local categorical minima (in accordance with Proposition 3.5), but has 300 unique local discrete minima. By sorting according to the *Increasing* and the *Central* procedure, the number of unique local discrete minima decreases to 168 and 90, respectively. Another measure of the "well-behavior" could be the average distance between discrete neighbors in the objective space, i.e.,

$$d_{\text{mean}} = \frac{\sum_{\{\mathbf{x},\mathbf{y}\in X:\|\mathbf{x}-\mathbf{y}\|=1\}} |f(\mathbf{x})-f(\mathbf{y})|}{\text{card}\{\mathbf{x},\mathbf{y}\in X : \|\mathbf{x}-\mathbf{y}\|=1\}}, \tag{15}$$

where the norm is Euclidean. For this measure, we have $d_{\text{mean}} = 0.32$ for the original problem, $d_{\text{mean}} = 2.26$ for the categorical problem (P), $d_{\text{mean}} = 1.09$ for the problem (P_Π) sorted with the *Increasing* procedure, and $d_{\text{mean}} = 0.86$ for the problem (P_Π) sorted with the *Central* procedure. The interpretation is that the discretely neighboring points are closer in terms of function values in the sorted domains compared to that with the arbitrary ordering, however not as close as for the original numerical problem.

We conclude, both by visual inspection of the figures and from the numeric well-behavior measures, that the two sorting procedures were able to reconstruct some of the structure of the original unknown problem. For this illustrative problem, the *Central* sorting procedure seems to be the most successful.



(a) Original problem.

(b) The categorical problem (P) given by a randomized ordering of the variables.

(c) The sorted problem using the *Increasing* procedure.

(d) The sorted problem using the *Central* procedure.

Figure 4: A discretized version of the Goldstein-Price test function. A lighter color means a larger function value than a darker. In (c) and (d) the obtained problems (P_{Π}) are obtained by construction of a sortings using the two different procedures. The sortings have been constructed at the (globally optimal) original point $(0, -1)^{\mathrm{T}}$, marked with a ring in all figures.

To summarize, we have in this section suggested a simple neighborhood search method in order to find categorical local minima. In order to find good local minima, a discrete global descent framework is utilized. Within the framework, the domain is repeatedly sorted with the aim of finding numerized problems that are as easy as possible to solve utilizing the numerical optimization technique provided by the global descent method.

5 Experimental results

We have implemented the global descent method from in [35] in MATLAB® ([28]); its principal steps are shown in Algorithm 2. To the method we have added the extensions of categorical local search and the different variants of sorting given by Algorithms 3 and 4.

Test problems from the literature on discrete global descent methods and global unconstrained optimization have been selected, and the method has also been applied to two versions of two real-world problems considering the configuration of heavy-duty trucks at Volvo 3P. For the (numerical) test examples from the literature, categorical counterparts have been created by constructing random orderings of the variables in each dimension. All problems have been solved with and without the extension of a categorical local search, and with and without the extension of the two sorting variants. To reduce the effects of randomness, all problems have been solved 50 times. In each run, the starting point has been selected randomly in the domain.

The results from the numerical experiments are shown in Table 1 for the test problems from the literature and in Table 5 for the real-world problems. The first three columns of the tables specify the problem, the neighborhood definition (*D = Discrete*, *C = Categorical*) and the sorting strategy (*N = No sorting*, *I = Increasing*, *C = Central*). The fourth and the fifth columns contain the approximation of the global minimum obtained and the proportion of runs finding a global minimum $\mathbf{x}^*_{\text{glob}}$. The sixth and the seventh columns contain, respectively, the number of unique discrete minima and the objective function distance d_{mean} as defined in (15) between discretely neighboring points at the final numerized problem sorted at the global minimizer obtained. The number of discrete minima has been estimated by generating $10{,}000$ uniformly distributed points in the domain, checking how many of these that are unique local minimizers, and extrapolating the result to the whole domain. In a similar way, pairs of points have been generated at random, and their objective function distances have been computed and extrapolated. The eighth column contains the number of main iterations of the global descent method. In the ninth and tenth columns of the tables, the number of unique function evaluations used, and, for the runs where the true optimum $\mathbf{x}^*_{\text{glob}}$ is found, the number of unique function evaluations until $\mathbf{x}^*_{\text{glob}}$ is found, are presented. The last column in the tables contains the proportion of all configurations in the domain that have been evaluated for each problem.

The test problems selected from the literature, with modifications, are the following:

Colville ([22, 35])

$$\begin{array}{rll} \text{minimize} & f(\mathbf{x}) := 100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2 & \\ & \quad +10.1\big((x_2-1)^2+(x_4-1)^2\big)+19.8(x_2-1)(x_4-1), & \\ \text{subject to} & x_i \in [-10,10], & i=1,\dots,4, \\ & x_i \quad \text{integer}, & i=1,\dots,4. \\ \text{Optimal solution:} & \mathbf{x}^*_{\text{glob}} = [1,1,1,1],\ f(\mathbf{x}^*_{\text{glob}}) = 0. & \end{array}$$

Powell's singular function ([30, 35]) (discretized)

$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) := (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^2, \\
\text{subject to} \quad & x_i = \frac{y_i}{10}, \qquad i = 1, \dots, 4, \\
& y_i \in [-100, 100], \qquad i = 1, \dots, 4, \\
& y_i \quad \text{integer}, \qquad i = 1, \dots, 4. \\
\text{Optimal solution:} \quad & \mathbf{x}^*_{\text{glob}} = [0, 0, 0, 0], \ f(\mathbf{x}^*_{\text{glob}}) = 0.
\end{aligned}$$

Shekel10 ([12]) (discretized)

$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) := -\sum_{j=1}^{10} \frac{1}{c_j + \sum_{i=1}^{4} (x_i - A_{ij})^2}, \\
\text{subject to} \quad & x_i \in [0, 10], \qquad i = 1, \dots, 4, \\
& x_i \quad \text{integer}, \qquad i = 1, \dots, 4, \\
\text{where} \quad & \mathbf{c} = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)^{\mathrm{T}}, \\
& \mathbf{A} = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{bmatrix}. \\
\text{Optimal solution:} \quad & \mathbf{x}^*_{\text{glob}} = [4, 4, 4, 4], \ f(\mathbf{x}^*_{\text{glob}}) \approx -10.536.
\end{aligned}$$

The real-world problems have been supplied by Volvo 3P and consider finding (feasible) truck configurations as in Example 1.1 that are optimal with respect to two measures, one related to the rollover stability, and one related to the transport efficiency of the truck. To create test problems of reasonable sizes, the configuration space in which to search has been limited to 6 dimensions by starting from an actually produced truck and by allowing changes in some design options that have been determined to be particularly important. These design options consider, e.g., the axle arrangement (7 alternatives), the battery box mounting (14 alternatives) and the wheel brake type (19 alternatives). In total, the configuration space consists of $312,816$ truck specifications. For each of the two objectives, two problem instances have been constructed: one variant considering the objective function exactly as given from a computer simulation, and one variant where a normally distributed deterministic noise has been added to the objective space in order to simulate some uncertainty in the objective functions as if they were, e.g., given by physical experiments. The latter variant should also increase the difficulty of the problem by increasing the number of local minima. The objective functions have in all cases been linearly scaled such that $f(\mathbf{x}) = 0$ for the globally optimal configuration and $f(\mathbf{x}) = 1$ for the worst configuration in the configuration space. Monte-Carlo simulation has been used in order to validate that the test problems fulfill the continuity-like properties *P1* and *P2*. The results indicate that all problems, both the ones from the literature and the real-world problems, fulfill both properties *P1* and *P2*.

Table 1: Experimental results when applying the different versions of discrete global descent to test examples selected from the literature.

Problem	N.	S.	$f(\mathbf{x}^*_{\text{final}})$	prop. runs $f(\mathbf{x}^*_{\text{glob}})$	#discr. min	d_{mean}	#glob. desc. its	#fcn ev. tot.	#fcn ev. unt. $\mathbf{x}^*_{\text{glob}}$	prop. ev. pts
Colville	*D*	*N*	36	0.24	3404	1.8e5	8.2	3,112	2,279	1.6e-2
	C	*N*	54	0.36	3433	1.8e5	2.2	2,154	1,123	1.1e-2
	C	*I*	183	0.18	473	4.8e4	1.6	1,668	440	8.6e-3
	C	*C*	0	1	57	5.7e4	3.4	2,046	1,068	1.1e-2
Powell	*D*	*N*	6.4	0	4.9e7	1.2e5	17.7	38,884	–	2.4e-5
	C	*N*	0.3	0.02	4.9e7	1.2e5	1.1	27,462	19,962	1.7e-5
	C	*I*	0	1	3.8e8	1.5e5	4.6	36,568	24,525	2.2e-5
	C	*C*	0	1	1	5.9e3	4.6	38,565	27,208	2.4e-5
Shekel10	*D*	*N*	-5.8	0.36	435	0.099	4.4	963	567	6.6e-2
	C	*N*	-5.4	0.26	436	0.097	1.3	660	202	4.5e-2
	C	*I*	-4.3	0.12	401	0.071	1.1	725	103	5.0e-2
	C	*C*	-9.4	0.78	127	0.064	2.1	912	384	6.2e-2

Table 2: Experimental results when applying the different versions of discrete global descent to the real-world problems.

Problem	N.	S.	$f(\mathbf{x}^*_{\text{final}})$	prop. runs $f(\mathbf{x}^*_{\text{glob}})$	#discr. min	d_{mean}	#glob. desc. its	#fcn ev. tot.	#fcn ev. unt. $\mathbf{x}^*_{\text{glob}}$	prop. ev. pts
Volvo1	*D*	*N*	0	1	844	0.088	4.66	3,593	2,854	1.1e-2
	C	*N*	0	1	818	0.086	1.04	1,513	161	4.8e-3
	C	*I*	0	1	783	0.063	1.10	1,662	175	5.3e-3
	C	*C*	0	1	267	0.065	1.06	2,193	161	7.0e-3
Volvo1b	*D*	*N*	8.4e-4	0.64	1,035	0.088	6.0	3,556	1,882	1.1e-2
	C	*N*	5.0e-4	0.76	1,023	0.090	1.4	2,526	587	8.1e-3
	C	*I*	0	1	877	0.062	1.7	2,143	622	6.9e-3
	C	*C*	4.9e-5	0.98	496	0.069	1.6	2,653	643	8.5e-3
Volvo2	*D*	*N*	0	1	404	0.092	2.26	2,924	672	9.3e-3
	C	*N*	0	1	405	0.094	1.02	2,479	114	7.9e-3
	C	*I*	0	1	482	0.059	1.02	1,490	112	4.7e-3
	C	*C*	0	1	202	0.078	1.04	1,922	113	6.1e-3
Volvo2b	*D*	*N*	2.9e-3	0.06	2,503	0.092	6.4	4,207	3,740	1.3e-2
	C	*N*	2.4e-3	0.10	2,359	0.090	2.4	3,392	226	1.0e-2
	C	*I*	4.6e-3	0.40	1,317	0.056	2.4	2,427	1,085	7.7e-3
	C	*C*	1.0e-3	0.30	877	0.075	2.8	3,459	1,123	1.1e-2

From the tables, we conclude that both our suggested extensions to the discrete global descent method seem to improve its performance in solving categorical optimization problems. The first extension, to use a categorical neighborhood based on the Hamming metric, results for all problems but one (**Shekel 10**) at a better (or equal) approximation of the global minimum. Additionally, in all these cases, the ratio of runs ending up in the global minimum $\mathbf{x}^*_{\text{glob}}$ is greater (or equal) when using the categorical neighborhood instead of the discrete one. We also note that the number of function evaluations needed until the global optimum is found, is often much smaller when using the categorical neighborhood.

For the different sorting strategies, the analysis is not as clear. For the test examples in Table 1 the *Central* strategy is clearly the most successful one with the best approximations of the global minimum, and the highest ratios of runs ending up at a true global minimum. However, the numbers of function evaluations to find the true global minima are lower for the *Increasing* and, when succeeding in finding them, also in some cases when not using sorting at all.

We see that the number of unique local minima is for all problems the lowest when using the *Central* sorting strategy. However, the number of local minima cannot always be used as an indicator of how easily solvable a problem is; see problem **Powell** with the *Increasing* sorting strategy compared to no sorting. The indicator given by d_{mean} is not always good either; see problem **Colville** where d_{mean} is the lowest for the *Increasing* strategy, but where this strategy leads to poor approximations of the global minimum and a low ratio of success in finding $\mathbf{x}^*_{\text{glob}}$.

Concerning the real-world problems, the first conclusion is that the method suggested works perfectly, but is overly sophisticated for these categorical problem instances. In every case the problems are solved to global optimality by just considering them as a nonlinear integer program with the given variable ordering, and by applying the standard discrete global descent method. We see, however, that using the categorical neighborhood definition dramatically reduces the number of function evaluations needed.

For the real-world problems with deterministic noise added to the objective functions, the results, as for the numerical test examples, indicate that both using categorical neighborhoods in the local optimizations, and using sortings when constructing the global descent functions, improve the performance when applying a discrete global descent method to categorical problems. It is, however, hard to say in this case if the *Increasing* or the *Central* sorting strategy works best. Clearly, more testing is needed.

6 Outlook

In this article we argue that pure categorical optimization problems must be studied and solved. We show that the discrete global descent method for nonlinear integer programming problems can be utilized in order to solve the categorical problems and we suggest mathematical properties that should hold in order for the proposed solution methods to work. The mathematical properties should be analyzed further. It is possible that there are other properties that better reflect whether a categorical optimization problem can be solved or not using our suggested technique. It is also possible that there are other, essentially different, properties that often are fulfilled for problems from certain applications, enabling other solution techniques to be exploited.

In the article, discrete global descent methods for categorical problems have been developed by converting the problems to (families of) numerical nonlinear integer programs. This evolution resembles the early development of extensions of global descent methods from continuous to numerically discrete problems where discrete problems were converted to approximate continuous counterparts. Later, descent methods explicitly concerning discrete functions were developed. It is an open question whether there is some global descent function that explicitly considers a categorical problem as the categorical problem it really is.

We deal in this article with infeasible points by considering them as feasible but by assigning a sufficiently large penalty value to them such that they will not be interesting for the descent method. This is reasonable if the constraints are of a general set type, where the only output is if a point is feasible or not. If a constraint evaluation gives more information than just yes / no, then some more intelligent constraint handling technique should be developed. A suitable technique depends, however, on the characteristics of the evaluation of the particular constraints in question.

It would be interesting to develop other methods for finding well-behaved numerical representations of the categorical problems than the ones based on sortings in the categorical neighborhood presented. It would also be interesting to develop and evaluate more measures or indicators of how well a numerical representation is behaved and how these measures or indicators correspond to the result when solving the numerized instances with numerical techniques. The measures suggested give a hint about the well-behavior; however, it is obvious that just counting the number of discrete local minima in the domain or measuring the difference in objective values for neighbors in the domain does not say everything about how easily a problem is solved to global optimality.

More numerical testing of the method suggested, and also comparisons against methods of essentially different types, clearly is needed in order to evaluate the performance of suggested method. This article serves as a proof of concept that discrete global descent could be used for categorical problems, and hence that such problems can be approached with mathematical techniques.

Acknowledgments

The authors wish to thank Volvo 3P, and especially Tomas Hammar, for supplying data for the real-world test problems.

References

[1] K. Abhishek, S. Leyffer, and J. T. Linderoth, *Modeling without categorical variables: a mixed-integer nonlinear program for the optimization of thermal insulation systems*, Optimization and Engineering, 11 (2010), pp. 185–212.

[2] M. A. ABRAMSON, *Pattern search algorithms for mixed variable general constrained optimization problems*, PhD thesis, Rice University, Houston, TX, USA, 2003.

[3] M. A. ABRAMSON, C. AUDET, J. W. CHRISSIS, AND J. G. WALSTON, *Mesh adaptive direct search algorithms for mixed variable optimization*, Optimization Letters, (2009), pp. 35–47.

[4] C. AUDET AND J. E. DENNIS JR., *Pattern search algorithms for mixed variable programming*, SIAM Journal on Optimization, 11 (2001), pp. 573–594.

[5] ———, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 17 (2006), pp. 188–217.

[6] J. BEAL, A. SHUKLA, O. BREZHNEVA, AND M. ABRAMSON, *Optimal sensor placement for enhancing sensitivity to a change in stiffness for structural health monitoring*, Optimization and Engineering, 9 (2008), pp. 119–142.

[7] D. R. BROWN AND K.-Y. HWANG, *Solving fixed configuration problems with genetic search*, Research in Engineering Design, 5 (1993), pp. 80–87.

[8] S. E. CARLSON, *Genetic algorithm attributes for component selection*, Research in Engineering Design, 8 (1996), pp. 33–51.

[9] S. E. CARLSON, R. SHONKWILERM, AND E. INGRIM, *Comparison of three non-derivative optimization methods with a genetic algorithm for component selection*, Journal of Engineering Design, 5 (1994), pp. 367–378.

[10] S. CARLSON-SKALAK, M. D. WHITE, AND Y. TENG, *Using an evolutionary algorithm for catalog design*, Research in Engineering Design, 10 (1998), pp. 63–83.

[11] M. W. COOPER, *A survey of methods for pure nonlinear integer programming*, Management Science, 27 (1981), pp. 353–361.

[12] L. C. W. DIXON AND G. P. SZEGÖ, eds., *Towards Global Optimization*, Amsterdam, Holland, 1975, Elsevier, North-Holland.

[13] P. FAVATI AND F. TARDELLA, *Convexity in nonlinear integer programming*, Ricerva Operativa, 53 (1990), pp. 3–44.

[14] M. FUCHS AND A. NEUMAIER, *Discrete search in design optimization*, in Proceedings of the First International Conference on Complex Systems Design & Management (CSDM), Paris, France, 2010, pp. 113–122.

[15] ———, *A splitting technique for discrete search based on convex relaxation*, Journal of Uncertain Systems, 4 (2010), pp. 14–21.

[16] S. FUJISHIGE AND K. MUROTA, *Notes on L-/M-convex functions and the separation theorems*, Mathematical Programming, 88 (2000), pp. 129–146.

[17] R. GE, *A filled function method for finding a global minimizer of a function of several variables*, Mathematical Programming, 46 (1990), pp. 191–204.

[18] R. GE AND C. HUANG, *A continuous approach to nonlinear integer programming*, Applied Mathematics and Computation, 34 (1989), pp. 39–60.

[19] A. A. GOLDSTEIN AND J. F. PRICE, *On descent from local minima*, Mathematics of Computation, 25 (1971), pp. 569–574.

[20] J. P. GONZALEZ-ZUGASTI, K. N. OTTO, AND J. D. BAKER, *A method for architecting product platforms*, Research in Engineering Design, 12 (2000), pp. 61–72.

[21] P. T. HELO, Q. L. XU, S. J. KYLLÖNEN, AND J. JIAO, *Integrated vehicle configuration system–connecting the domains of mass customization*, Computers in Industry, 61 (2010), pp. 44–52.

[22] W. HOCK AND K. SCHITTKOWSKI, *Test examples for nonlinear programming codes*, Journal of Optimization Theory and Applications, 30 (1980), pp. 127–129.

[23] J. JIAO, T. W. SIMPSON, AND Z. SIDDIQUE, *Product family design and platform-based product development: a state-of-the-art review*, Journal of Intelligent Manufacturing, 18 (2007), pp. 5–29.

[24] J. JIAO, Y. ZHANG, AND Y. WANG, *A generic genetic algorithm for product family design*, Journal of Intelligent Manufacturing, 18 (2007), pp. 233–247.

[25] M. KOKKOLARAS, C. AUDET, AND J. E. DENNIS JR., *Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system*, Optimization and Engineering, 2 (2001), pp. 5–29.

[26] J. LE BESNERAIS, A. FASQUELLE, V. LANFRANCHI, M. HECQUET, AND P. BROCHET, *Mixed-variable optimal design of induction motors including efficiency, noise and thermal criteria*, Optimization and Engineering, 12 (2011), pp. 55–72.

[27] S. LUCIDI, V. PICCIALLI, AND M. SCIANDRONE, *An algorithm model for mixed variable programming*, SIAM Journal on Optimization, 15 (2005), pp. 1057–1084.

[28] MATLAB, version 7.5.0 (R2007b). The Mathworks Inc., Natick, MA, USA, `www.mathworks.se/products`.

[29] B. L. MILLER, *On minimizing nonseparable functions defined on the integers with an inventory application*, SIAM Journal on Applied Mathematics, 21 (1971), pp. 166–185.

[30] J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, 7 (1981), pp. 17–41.

[31] K. MUROTA, *Discrete convex analysis*, Mathmatical Programming, 83 (1998), pp. 313–371.

[32] ———, *Discrete Convex Analysis: Monographs on Discrete Mathematics and Applications 10*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[33] K. MUROTA AND A. SHIOURA, *M-convex function on generalized polymatroid*, Mathematics of Operations Research, 24 (1999), pp. 95–105.

[34] K. MUROTA AND A. SHIOURA, *Relationship of M-/L-convex functions with discrete convex functions by Miller and Favati–Tardella*, Discrete Applied Mathematics, 115 (2001), pp. 151–176.

[35] C.-K. NG, D. LI, AND L.-S. ZHANG, *Discrete global descent method for discrete global optimization and nonlinear integer programming*, Journal of Global Optimization, 37 (2007), pp. 357–379.

[36] C.-K. NG, L.-S. ZHANG, D. LI, AND W.-W. TIAN, *Discrete filled function method for discrete global optimization*, Computational Optimization and Applications, 31 (2005), pp. 87–115.

[37] N. SENIN, D. R. WALLACE, N. BORLAND, AND M. J. JAKIELA, *Distributed modeling and optimization of mixed variable design problems*, Technical report, CADlab, Department of Mechanical Engineering, MIT, Cambridge, MA, USA, 1999.

[38] Y.-L. SHANG AND L.-S. ZHANG, *Finding discrete global minima with a filled function for integer programming*, European Journal of Operational Research, 189 (2008), pp. 31–40.

[39] T. W. SIMPSON, *Product platform design and customization: Status and promise*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 18 (2004), pp. 3–20.

[40] T. W. SIMPSON, J. R. MAIER, AND F. MISTREE, *Product platform design: method and application*, Research in Engineering Design, 13 (2001), pp. 2–22.

[41] T. A. Sriver, J. W. Chrissis, and M. A. Abramson, *Pattern search ranking and selection algorithms for mixed variable simulation-based optimization*, European Journal of Operational Research, 198 (2009), pp. 878–890.

[42] J. G. Walston, *Search Techniques for Multi-Objective Optimization of Mixed-Variable Systems Having Stochastic Responses*, PhD thesis, Graduate School of Engineering and Management, Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, OH, USA, 2007.

[43] S. F. Woon and V. Rehbock, *A critical review of discrete filled function methods in solving nonlinear discrete optimization problems*, Applied Mathematics and Computation, 217 (2010), pp. 25–41.

[44] Y. Yang, Z. Wu, and F. Bai, *A filled function method for constrained nonlinear integer programming*, Journal of Industrial and Management Optimization, 4 (2008), pp. 353–362.

[45] Y. Yang and L. Yumei, *A new discrete filled function algorithm for discrete global optimization*, Journal of Computational and Applied Mathematics, 202 (2007), pp. 280–291.

[46] Z. Zhao, J. C. Meza, and M. V. Hove, *Using pattern search methods for surface structure determination of nanomaterials*, Journal of Physics: Condensed Matter, 18 (2006), pp. 8693–8706.

[47] W. X. Zhu, *An approximate algorithm for nonlinear integer programming*, Applied Mathematics and Computation, 93 (1998), pp. 183–193.