Advanced Engineering Informatics 28 (2014) 11–27

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Supporting the design process with hypergraph genetic operators

B. Strug *, E. Grabska, G. Ślusarczyk

Department of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Reymonta 4, 30-059 Kraków, Poland

ARTICLE INFO

Article history:
Received 14 October 2012
Received in revised form 28 October 2013
Accepted 30 October 2013
Available online 27 November 2013

Keywords:
Hypergraph representation
Genetic operators
Evolutionary design
CAD

ABSTRACT

In this paper an evolutionary technique is proposed as a method for generating new design solutions with genotypes represented in the form of hierarchical hypergraphs. Such hypergraph-based evolutionary design requires the adaptation of traditional evolutionary operators like cross-over and mutation. This paper presents an attempt at defining modified cross-over and mutation operators that act on hierarchical hypergraphs. The application of the proposed transformations is illustrated on examples of designing floor layouts and furniture with the use of the evolutionary method.

1. Introduction

This paper deals with graph-based evolutionary design. In the proposed approach a graph-based representation of genotypes is used. It enables to express geometrical properties of objects together with their attributes and relations between object components. The internal representation of designed individuals in the form of hierarchical hypergraphs allows for expressing both multi-argument relations of different types (like coplanarity) and hierarchical dependencies between object parts, which are impossible to express using other structures.

The used hierarchical hypergraphs can represent any artefact being designed at different levels of detail and at different stages of the design process, thus hiding unnecessary at the moment low-level data and presenting the designer only an outline of the object or showing a detailed view of the whole object (or of its part) [16]. Hypergraph hierarchy reflects the top-down way of the design solution development and allows for performing partial modifications of solutions without changing the whole graph structure. Hierarchical hypergraphs encode the knowledge about created designs and therefore enable to efficiently reason about the conformity of designs with specified design criteria [18]. Moreover, the hypergraph hierarchy facilitates the reasoning process by giving the possibility of selecting both the hierarchy level and the appropriate part of the solution.

Designing new artifacts requires methods of generating hierarchical hypergraphs representing them and of modifying them to change/improve the represented objects. A sequence of such modifications is expected to find the best object in terms of the predefined target function. Thus such a process can be considered as a search process within a space of hierarchical hypergraphs and/or objects represented by them.

The use of hierarchical hypergraphs as the representation of genotypes in an evolutionary search requires the adaptation of traditional evolutionary operators like cross-over and mutation. As the hypergraphs selected to be transformed by the evolutionary operators and their structures are not known a priori, the operators must allow for a dynamic computation of new hypergraphs. Thus the operators have to be specified by an algorithm rather than a set of rules.

In this paper modified genetic operators are specified for hypergraphs. The notion of hypergraph homology is proposed to facilitate selection of subgraphs to be exchanged by cross-over [40]. Three different ways of defining such a homology are discussed. Homologous subgraphs represent parts of design objects playing the same functional role but having different internal structures. Homology effectively guides the recombination of solutions as it enables the evolutionary search mechanism to have some 'understanding' of the solutions meaning. Mutation affects local and global attributes as well as hypergraph structures (by adding or deleting subgraphs).

The presented approach is illustrated by examples of designing floor layouts and furniture, with their genotypes in the form of hierarchical hypergraphs. Examples concerning floor layouts come from the Hypergraph System Supporting Design and Reasoning (HSSDR system) [17], while examples presenting furniture are obtained using the Furniture Design application [39]. Both programs are written in Java language.

* Corresponding author. Tel.: +48 126635547.
E-mail addresses: barbara.strug@uj.edu.pl (B. Strug), uigrabsk@cyf-kr.edu.pl (E. Grabska), gslusarc@uj.edu.pl (G. Ślusarczyk).

1474-0346/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.aei.2013.10.002

The paper is structured in the following way. The next section describes the related work. In Section 3 the model of the design process encompassing the evolutionary method is presented. In Section 4 the attributed hierarchical hypergraphs used in this paper are discussed. In Section 5 the phases of evolutionary design, in which hierarchical hypergraphs play an important role, are described. Evolutionary hypergraph-based operators are proposed and examples of application of the presented method are described. In Section 6 some problems related to this approach are discussed. Finally, in the last section advantages and disadvantages of this approach as well as possible future research directions are briefly described.

2. Related work

This paper deals with issues from different domains. Hence in the review of related work several domains are taken into account, such as methods of object representation in design, especially structural ones, graph-based representations, different approaches to graph generation, different types of graphs that can be used for design representation, but also different types of evolutionary algorithms, especially those attempting to use a non-linear representation of solutions and the ones applied to solving design tasks.

From the ontological point of view relevant aspects for designing new artifacts include quantitative spatial constraints, qualitative relations, functionally-dependent and abstract conceptualizations [6]. The approach in the context of an industrial standard for data representation and interchange in the architectural domain, named the Industry Foundation Classes is presented in [20].

A number of different approaches have been used to represent objects in computer aided design systems, such as boundary representations, sweep-volume representations, surface representations or CSG (constructive solid geometry) [22,26,27]. These methods rely mainly on geometrical features of an object, like the size, position of its elements but do not take into account structural interconnections among object parts, like being adjacent or being part of other elements. In order to represent both relations and geometrical features graphs have been proposed [14]. They have been used in numerous application domains to represent the structure and topology of different objects. One of the first graph representations has been based on the boundary representation, where a so called face adjacency graph was used to represent an object consisting of faces [2]. Another domain is related to shape representation and recognition, where attributed relational graphs have been used, with nodes representing primitives (usually in a form of vectors and quadrilaterals) and edges describing relations between primitives [7,10,34].

The problem of generating new designs can be seen as a search problem, which is a well established domain in computer science [30]. One of the search methods is the evolutionary technique. It is based on natural evolution. Instead of one solution at a time, a larger subset of the search space, known as a population, is considered. Hence, instead of one design a large number of them can be evaluated, tested and refined at the same time. Moreover, as evolutionary search consists in evaluating and refining possible solutions it can be seen as analogous to a human design iterative process of analysis, testing and optimization [5]. Similarly to the refinement step in human design, which is based on earlier analysis and testing, in evolutionary search designs to be transformed are determined according to their evaluation (so called fitness). The fitter the design the more chances it has to contribute to the newly generated, refined designs [12,21].

There is no definition of an evolutionary algorithm but it is usually agreed that some standard elements must be provided: a population of solutions, a method of coding them, genetic operators that are capable of generating new elements, an evaluation method, a method of selecting elements to be transformed and a stop condition for the whole process [5,12,21,30]. There has also been a lot of research in different types of evolutionary algorithms. The domain collectively called evolutionary computation deals with different types of approaches. They include such approaches as genetic algorithms (GA) introduced by Holland [21], evolutionary strategies (ES), evolutionary programming (EP) and genetic programming (GP). A comparison of the first three types of evolutionary algorithms was presented by Bäck and Schwefel [3]. A good overview of both theoretical aspects and practical applications of evolutionary computations was presented by Bäck [4]. As evolutionary algorithms are usually used as an optimization tool there has also been a lot of work on using them in multi-objective optimization [11,44–46].

Evolutionary computation has been used in solving numerous problems in many domains of computing. The so called mechanism design, which consists in finding rules for achieving a specified goal is an example of such domains [33]. Evolutionary approach has also been used in game design [8].

In the domain of computer aided design evolutionary methods were used in different fields. An approach which uses evolutionary computation in a multi-agent design environment was proposed by Liu and Tang [25], who implemented the evolutionary algorithm which helps designers in creative mobile phone design. As a representation of the mobile phone shape a binary algebraic expression tree is used and a feature based product tree is then used to represent component combination choices, and a human is expected to select a design. Thus this work uses some structural information about designed objects in a form of a tree, but the authors do not attempt to design a new shape, but rather to combine elements of the existing ones. An evolutionary approach to solving design tasks in an interactive design environment based on the framework which uses shape grammars to generate design has been proposed by Lee and Tang et al. [24]. The evolutionary system has also been used to generate 3D objects by Clune and Lipson [9], but they used a geometrical representation of objects rather than a structural one. Another example of interactive evolutionary design of 3D objects using graph grammars, where graph nodes were labelled by Euclidean coordinates, was presented at MusArt conference by McDermott [28]. In his approach the actual generation was run by the user and consisted in applying subsequent rules, thus making it more similar to grammatical derivation than to the evolutionary process. Even though the selection and user-based fitness function were used, in place of evolutionary operators the rewriting process was carried out. A collaborative use of the evolutionary approach has been also researched as a mean to analyse a design space [36].

There has not been much research on using graphs as a basis for evolutionary search. Sims [37,38] used graphs to represent 3D structures, but the operations where performed at the file level, i.e., the structures were serialized to files, and then treated as strings within the evolutionary algorithm. Thus the structural information describing relations was not taken into account. Some research has also been done on using the hierarchical, tree-like chromosome in Finite Element Method (FEM) optimization [32]. The operators defined in those approaches cannot be used for graph structures because relations in hierarchical hypergraphs are much more complex, so they would not preserve the whole structural information [31].

Thus the main contribution of this paper is a formal introduction of genetic operators applied to hierarchical hypergraphs. Moreover three levels of homology were proposed to allow for selecting similar subgraphs. The method of exchanging