Composing counterpoint musical scores with variable neighborhood search

D. Herremans and K. Sörensen

University of Antwerp, Faculty of Applied Economics,
Operations Research Group ANT/OR,
{dorien.herremans,kenneth.sorensen}@ua.ac.be

Abstract

In this research a variable neighborhood search (VNS) algorithm is developed that can generate musical fragments of arbitrary length. The objective function of the algorithm is based on a quantification of existing fifth species counterpoint rules. A thorough parametric analysis of the VNS reveals the significance of the algorithm's parameters on the quality of the composed fragment, as well as their optimal settings. The VNS algorithm has been implemented in a user-friendly software environment for composition, called *Optimuse*.

Keywords: Variable neighborhood search, Metaheuristic, Computer Aided Composing

Introduction

In this paper a variable neighborhood search (VNS) algorithm is developed that can generate fifth species counterpoint fragments. Counterpoint is a specific type of polyphonic classical music that consists of two parts or melodies: a *cantus firmus* and a *counterpoint* melody. The cantus firmus is a base melody to which the counterpoint melody is added.

Computer aided composing is a relatively new research area that focuses on using computers to assist composers. In its most extreme form the computer generates the entire musical piece. This can be done by realizing that composing music can—at least partially—be regarded as a *combinatorial optimization problem*, in which one or more melodies are "optimized" to fit the "rules" of their specific musical style. The rules written by Johann Fux in 1725 for the species counterpoint style are one of the most restrictive sets of rules for composing classical music [2], this makes them ideal to use as quantifiers of quality in an optimization context. We have quantified these rules, so that each rule receives a subscore between 0 and 1. The weighted summation over all the subscores gives us the objective function. The algorithm uses this score to determine how well an automatically composed fragment fits the counterpoint style.

VNS

In a previous paper, we have developed a VNS that can generate cantus firmus and *first* species counterpoint fragments. The VNS algorithm that is developed in this research is based on this previous algorithm and composes *fifth* species, a more complex form of counterpoint that has an added rhythmic aspect.

A fragment that has the correct number of notes, all of which are in the specified key, and that respects all the rhythmic criteria, is called *feasible*. The functioning of the VNS algorithm ensures that no infeasible fragments are generated. The initial start fragment is generated by a set of rules to ensure that all the rhythmic criteria are fulfilled.

Three different move types are defined, to form the different neighborhoods that the algorithm uses. The first move type swaps the place of a pair of notes (swap). The change1 move changes the pitch of any one note to any other allowed pitch. The last move, change2, is an extension of the previous one whereby the pitch of two sequential notes is changed simultaneously to all possible allowed pitches. The VNS performs a *tabu search* in each of the neighborhoods. When no improving solution can be found in any of the neighborhoods, a *perturbation* strategy allows the search to continue out of the local optimum [1]. The perturbation is implemented by reverting back to the best fragment and changing a predefined percentage of the notes to a new, random note from the key.

Often, the current fragment scores optimally with respect to a large majority of subscores, but performs badly with respect to some others. The VNS therefore also uses an *adaptive weight adjustment* mechanism in order to steer the optimization in the right direction. This mechanism adapts the weights of the subscores of the objective function at the same time a perturbation is performed. The adaptive weight mechanism works by increasing the weight of the subscore that has the worst value by 1. The algorithm then uses the score based on these new weights to determine the quality of fragments in the neighborhoods. In order to determine whether a fragment is considered as the new current best, however, the algorithm always considers the original weights.

Several of the algorithm's parameters can be set in different ways. In order to find the optimal settings, a Multi-Way ANOVA (Analysis of Variance) with interaction effects was done, using the statistical software R. A thorough analysis of this model, combined with the mean plots, reveals the significance of the algorithm's parameters on the quality of the composed fragment and the running time, as well as their optimal settings.

The algorithm has been implemented in C++ as *Optimuse*, a plugin for the user-friendly software environment for composition *Musescore*.

References

[1] N. Mladenovic and P. Hansen, *Variable neighborhood search*, Computers & Operations Research, Elsevier, 24(11):1097–1100, 1997.

[2] F. Salzer and C. Schachter, *Counterpoint in composition: the study of voice leading*, Columbia University Press, 1969.