

GMTH Proceedings 2017

herausgegeben von
Florian Edler und Markus Neuwirth

Populäre Musik und ihre Theorien. Begegnungen – Perspektivwechsel – Transfers

17. Jahreskongress der
Gesellschaft für Musiktheorie
und

27. Arbeitstagung der
Gesellschaft für Populärmusikforschung
Graz 2017

herausgegeben von
Christian Utz



Dieser Text erscheint im Open Access und ist lizenziert unter einer
Creative Commons Namensnennung 4.0 International Lizenz.

This is an open access article licensed under a
Creative Commons Attribution 4.0 International License.

Pierre Funck

Die Mouton-Maschine

Ein algorithmischer Blick auf *stacked canons* in der Oberquart

ABSTRACT: Bei der vierstimmigen Chanson *En venant de Lyon* von Jean Mouton ist nur die unterste Stimme notiert; die restlichen drei Stimmen ergeben sich durch einen sukzessiven Strettakanon (einen sogenannten *stacked canon*) in der Oberquart im Semibrevis-Abstand. Die Dux-Fortschreitungsregeln für diesen Kanontyp (und andere Engführungskanons mit beliebiger Stimmenzahl und beliebigen Einsatzintervallen) können algorithmisch hergeleitet werden. Anhand dieser Fortschreitungsregeln kann ein anderer Algorithmus mit Hilfe eines Zufallsgenerators Kanons erzeugen. In diesem Aufsatz wird der Frage nachgegangen, ob man die für *stacked canons* in der Oberquart algorithmisch generierten Fortschreitungsregeln zu einem für die Improvisation praxistauglichen Regelsystem reduzieren kann. Außerdem wird untersucht, inwiefern *stacked canons* in der Oberquart mit mehr als vier Stimmen möglich und ästhetisch sinnvoll sind.

In the four-part chanson *En venant de Lyon* by Jean Mouton, only the lowest part is notated; the other three parts arise via a so-called stacked canon, with each successive voice entering a semibreve later and a fourth above. The progression rules for the dux voice in this canon type (and in other stretto canons with any number of parts and any interval between the parts) can be derived by an algorithm. Using these progression rules, another algorithm can produce randomly generated canons. For stacked canons at the upper fourth, this article investigates whether one can streamline the algorithmically generated rules into practical rules suitable for improvisation. Furthermore it is examined whether it is possible to generate aesthetically pleasing stacked canons at the fourth with more than four voices.

Schlagworte/Keywords: algorithmic composition; algorithmische Komposition; canon; En venant de Lyon; improvised counterpoint; improvisierter Kontrapunkt; Jean Mouton; Kanon; Renaissance counterpoint; Renaissance-Kontrapunkt; stacked canon; Strettakanon; stretto canon

Einleitung

Andrea Anticos Sammlung *Motetti noui & chanzoni franciose a quatro sopra doi* (Venedig 1520, Reprint Genève: Minkoff 1982) beginnt mit der vierstimmigen Chanson *En venant de Lyon* von Jean Mouton, bei der nur die Bassstimme notiert ist (Bsp. 1).

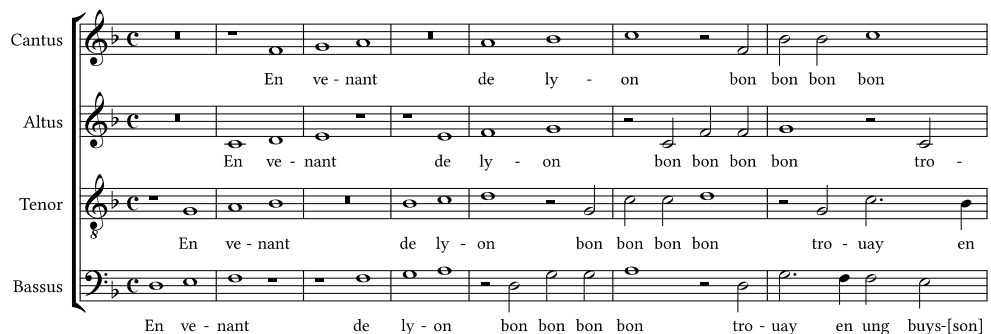
Joannes mouton 2



En venant de Lyon bon bon bon bon trou uay en fig buys[ro]b; et mari[us]
 il luy leuoit son pelisson ton ton bon bon; mais ie ne scay quilz font fo
 bin adit a marien bon bon ton bon uay bien garde uay bien garde mou
 to 1.

Beispiel 1: Jean Mouton, *En venant de Lyon* (in: *Motetti noui & chanzoni francoise a quatro sopra doi*, hg. von Andrea Antico, Venedig 1520)

Die restlichen drei Stimmen ergeben sich – wie die *signa congruentiae* suggerieren – aus einem sukzessiven Quartkanon im Semibrevis-Abstand. Eine Transkription der ersten sieben Mensuren findet man in Beispiel 2.



Cantus
 En ve - nant de ly - on bon bon bon bon

Altus
 En ve - nant de ly - on bon bon bon bon tro -

Tenor
 En ve - nant de ly - on bon bon bon bon tro - uay en

Bassus
 En ve - nant de ly - on bon bon bon bon tro - uay en ung buys-[son]

Beispiel 2: Mouton, *En venant de Lyon*, Transkription des Anfangs.¹

Kanons dieses Typs werden als Engführungskanons oder Strettakanons bezeichnet. Kompositionstechniken zur Erstellung solcher Kanons sind in der Literatur ausführlich besprochen worden.² Sind die Intervalle zwischen Dux und Comites

1 Nach Urquhart 2015; vollständige Transkription bei MacCracken/Winn 2014, 204, und auf <http://lmhs.oicrm.org/ressources/chansons-motetz/editions-modernes/chansons-et-motetz-de-pierre-attaingnant>. Für eine Diskussion der Probleme dieser Transkription siehe Urquhart 2015.

2 Prey 2012, 19–24 und 89–119, Ott 2014, 9–44, 131–151 und 169–177, Gauldin 1996, Urquhart 1997, Janin 2012, 81–84.

wie beim Mouton-Kanon alle gleich groß, spricht man auch von einem *stacked canon*.³

Stefan Prey und Immanuel Ott leiten für mehrere Einsatzschemata (d. h. Intervalle zwischen Dux und Comites) die Dux-Fortschreitungsregeln her – Ott ›von Hand‹ und Prey teils ›von Hand‹, teils mit Hilfe eines Algorithmus, der einen Graphen generiert, aus dem hervorgeht, welches melodische Dux-Intervall auf welches folgen kann.⁴ Dieser Graph, den wir in der Folge als ›Prey-Graphen‹ bezeichnen wollen, kann dann in einen weiteren Algorithmus eingespeist werden, der via Zufallsgenerator einen entsprechenden Kanon generiert.

Der vorliegende Aufsatz ist gleichsam eine Replik auf eine Fußnote von Prey:

Dieses und die folgenden Regelsysteme habe ich durch Probieren herausgefunden, mit Hilfe der Graphen und der Intervallkombinationstabellen. Es ist mir bisher nicht gelungen, ein systematisches Verfahren zu finden. Es ist auch unklar, ob das nützlich wäre; denn selbst wenn es gelänge, eine allgemeine Form eines Kanonregelsystems zu definieren, und ein Verfahren zu finden, das zu jedem Einsatzschema ein passendes Regelsystem generiert, so wäre immer noch unklar, ob ein solches Regelsystem so überschaubar wäre, dass es sich anwenden ließe.⁵

Der hier vorgestellte Algorithmus kann für ein gegebenes Einsatzschema ebenfalls Dux-Fortschreitungsregeln herleiten, benutzt aber einen anderen Ansatz und generiert keinen Graphen, sondern eine Liste von möglichen Intervallfolgen, mit Hilfe derer ebenfalls entsprechende Kanons via Zufallsgenerator erzeugt werden können. Die auf diese Weise algorithmisch generierten Regelsysteme und Kanons erlauben es, sich mit folgenden Fragen zu befassen:

- Kann man die Kanonregeln für drei- und mehrstimmige Oberquartkanons derart formulieren, dass sie in Echtzeit (z. B. in einer Live-Musiziersituation) systematisch angewendet werden können? Wenn nein, kann man brauchbare ›Untermengen‹ extrahieren, die in der Improvisationspraxis verwendbar sind?
- Inwiefern hat Mouton die Fortschreitungsregeln eines vierstimmigen Oberquartkanons ausgeschöpft?
- Bis zu welcher Stimmenzahl sind *stacked canons* in der Oberquart überhaupt möglich und ästhetisch sinnvoll?

3 Gosman 1997, Burn 2001. Für eine fundierte Kritik an Gosmans Ansatz siehe Ott 2014, 171–177.

4 Prey 2012, 89–119, Ott 2014, 276–280.

5 Prey 2012, 93.

Arbeitshypothesen

Vier Arbeitshypothesen bzw. Einschränkungen werden hier sowohl der ›händischen‹ als auch der algorithmischen Kanonregel-Herleitung zugrundegelegt:

1. *Note-gegen-Note-Satz*. Diese Hypothese ist zugegebenermaßen der programmiertechnischen Einfachheit geschuldet. Der Autor ist sich dessen bewusst, dass es bei Engführungskanons Passagen geben kann, die sich nicht auf einen Note-gegen-Note-Satz reduzieren lassen (siehe unten).
2. *Diatonischer Tonraum*. Wie man an der Realisierung des Mouton-Kanons sieht, übernehmen die Comites lediglich die Anzahl der diatonischen Stufen innerhalb eines Dux-Intervalls, aber nicht die realen Intervallabstände – so kann z. B. eine große Terz beim Dux durchaus zu einer kleinen Terz bei einem Comes werden.⁶ Deshalb wird hier von einem siebenstufigen, diatonischen Tonraum ausgegangen. Die Korrektur von horizontalen und vertikalen *Mi-contrafa*-Problemen durch Akzidentien wird gleichsam den Interpretinnen und Interpreten überlassen. Dabei wird kaltblütig das Risiko in Kauf genommen, dass einige dieser *Mi-contrafa*-Probleme sich als unlösbar herausstellen können.
3. *Sprünge maximal eine Quinte aufwärts oder abwärts*. Der Grund für diese Einschränkung ist rein ästhetischer Natur. Größere Sprünge wären im Algorithmus problemlos implementierbar, aber die generierten Kanons werden bei größeren Sprüngen schnell ästhetisch unbefriedigend.
4. *Keine Pausen*. Der Grund für diese – musikalisch überhaupt nicht sinnvolle – Einschränkung besteht vor allem darin, dass das Hauptaugenmerk auf die Situation gerichtet ist, bei der alle Stimmen ›in Betrieb‹ sind.

Ein Algorithmus zur Bestimmung der möglichen Intervallfortschreitungen

Schauen wir uns zuerst ein triviales Beispiel an, nämlich den zweistimmigen Oberquartkanon. Damit der Comes konsonant mit dem Dux ist, sind für letzteren nur die fünf Intervallfortschreitungen 5 \searrow , 3 \searrow , 2 \searrow , 2 \nearrow und 4 \nearrow erlaubt. Allerdings sind die Intervallfolgen 5 \searrow 5 \searrow , 2 \searrow 2 \searrow und 4 \nearrow 4 \nearrow nicht möglich, da diese zu Oktav-

6 Vgl. Berger 1987, 155–159, Ott 2014, 10.

Quint- oder Einklangparallelen zwischen Dux und Comes führen würden.⁷ Somit sind nur die folgenden Intervallfolgen möglich:

	2↗ 4↗	2↘ 4↗	3↘ 4↗	5↘ 4↗
4↗ 2↗	2↗ 2↗	2↘ 2↗	3↘ 2↗	5↘ 2↗
4↗ 2↘	2↗ 2↘		3↘ 2↘	5↘ 2↘
4↗ 3↘	2↗ 3↘	2↘ 3↘	3↘ 3↘	5↘ 3↘
4↗ 5↘	2↗ 5↘	2↘ 5↘	3↘ 5↘	

Tabelle 1: Intervallfortschreitungen für den zweistimmigen Oberquartkanon

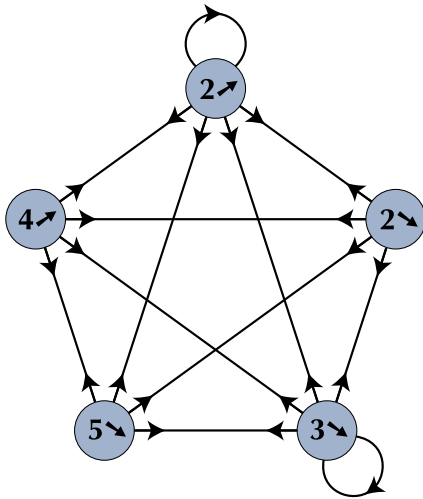


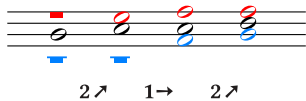
Abbildung 1: Prey-Graph für den zweistimmigen Oberquartkanon

Für einen zweistimmigen Kanon muss man also bereits Folgen von zwei Intervallen auf Konsonanz/Dissonanz und Parallelen überprüfen. Allgemein gilt: Für einen n -stimmigen Engführungskanon muss man alle Folgen von n Intervallen auf Konsonanz/Dissonanz und Parallelen überprüfen.

Genau dies ist die Vorgehensweise des *Brute-force*-Algorithmus, der hier eingesetzt wird. Von den neun in Betracht gezogenen Dux-Intervallen ($5↘ 4↘ 3↘ 2↘ 1↘ 2↗ 3↗ 4↗ 5↗$) werden zuerst sämtliche 9^n Kombinationen von Intervallen, also Intervallfolgen, generiert. Aus diesen Intervallfolgen werden dann diejenigen eliminiert, bei denen Dissonanzen oder Parallelen zwischen den Stimmen entstehen.

⁷ Antiparallelen, wie sie etwa bei $4↗ 5↘$ entstehen, werden toleriert, da sie auch im Mouton-Kanon vorkommen.

Die Versuchung ist groß, sich auf die Intervalle zu beschränken, die für den betreffenden zweistimmigen Kanon erlaubt sind (im Falle eines Oberquartkanons also $5 \searrow 3 \searrow 2 \searrow 2 \nearrow 4 \nearrow$). Es gibt aber Situationen, bei denen zum Beispiel der zweite Comes eine dissonante Quarte zwischen Dux und erstem Comes konsonant machen kann, etwa im Oberquart-Unterquint-Kanon (Bsp. 3).



Beispiel 3: Oberquart-Unterquint-Kanon

Hier darf der Dux also auf einem Ton stehenbleiben, obwohl das in einem reinen Quartkanon nicht möglich ist. Deshalb ist es sinnvoll, dass der Algorithmus *alle* Intervalle in Betracht zieht.

Nach der Dissonanz- und Parallelenüberprüfung bleibt für einen n -stimmigen Kanon eine – zuweilen umfangreiche – Menge von erlaubten Folgen von n Intervallen übrig. Jede Dux-Melodie, bei denen alle Ausschnitte von n sukzessiven Intervallen jeweils einer erlaubten Intervallfolge entsprechen, ergibt einen korrekten Kanon.

Implementiert wurde dieser Algorithmus (und der weiter unten besprochene Algorithmus, der Kanons generiert) in der Programmiersprache LISP, die sich trotz ihres hohen Alters – es handelt sich um die zweitälteste Programmiersprache nach Fortran – bis heute bei der algorithmischen Komposition großer Beliebtheit erfreut, da sie sehr leistungsstark im Umgang mit Listen (sprich: Notenfolgen, Intervallfolgen etc.) ist.⁸ Als Programmierumgebung wurde die Software Opusmodus verwendet.⁹

Der dreistimmige Oberquartkanon

Die vom oben beschriebenen Algorithmus generierte Intervallfolgenliste für den dreistimmigen Oberquartkanon ist vorerst unübersichtlich und wohl kaum praktisch einsetzbar (Tab. 2).

⁸ Der Autor ist Philippe Kocher (Zürcher Hochschule der Künste, Institute for Computer Music and Sound Technology) zu ewigem Dank verpflichtet für seine Einführung in LISP mit der Inbrunst des wahren Liebhabers und für die geduldige Unterstützung bei der Programmierarbeit.

⁹ <https://opusmodus.com>. Der LISP-Quellcode kann heruntergeladen werden unter https://storage.gmth.de/proceedings/articles/42/attachments/funck_mouton-maschine.zip

4↗ 2↘ 4↗	2↗ 4↗ 2↗	2↘ 4↗ 2↗	3↘ 4↗ 2↗	5↘ 4↗ 2↗
4↗ 2↗ 2↗	2↗ 4↗ 2↘	2↘ 4↗ 3↘	3↘ 4↗ 2↘	5↘ 4↗ 2↘
4↗ 2↗ 3↘	2↗ 4↗ 3↘	2↘ 4↗ 5↘	3↘ 4↗ 3↘	5↘ 4↗ 3↘
4↗ 2↗ 5↘	2↗ 4↗ 5↘	2↘ 3↘ 4↗	3↘ 4↗ 5↘	5↘ 2↗ 4↗
4↗ 2↘ 3↘	2↗ 2↗ 4↗	2↘ 3↘ 2↗	3↘ 2↗ 4↗	5↘ 2↗ 2↗
4↗ 2↘ 5↘	2↗ 2↗ 3↘	2↘ 3↘ 2↘	3↘ 2↗ 2↗	5↘ 2↗ 3↘
4↗ 3↘ 4↗	2↗ 2↗ 5↘	2↘ 3↘ 5↘	3↘ 2↗ 5↘	5↘ 2↗ 5↘
4↗ 3↘ 2↗	2↗ 3↘ 4↗	2↘ 5↘ 4↗	3↘ 2↘ 4↗	5↘ 2↘ 4↗
4↗ 3↘ 2↘	2↗ 3↘ 2↘	2↘ 5↘ 2↗	3↘ 2↘ 3↘	5↘ 2↘ 3↘
4↗ 3↘ 5↘	2↗ 3↘ 5↘	2↘ 5↘ 3↘	3↘ 2↘ 5↘	5↘ 3↘ 4↗
4↗ 5↘ 2↗	2↗ 5↘ 4↗		3↘ 5↘ 4↗	5↘ 3↘ 2↗
4↗ 5↘ 2↘	2↗ 5↘ 2↗		3↘ 5↘ 2↗	5↘ 3↘ 2↘
4↗ 5↘ 3↘	2↗ 5↘ 2↘		3↘ 5↘ 2↘	5↘ 3↘ 5↘
	2↗ 5↘ 3↘		3↘ 5↘ 3↘	

Tabelle 2: Intervallfortschreitungen für den dreistimmigen Oberquartkanon

Preys Algorithmus `>kgra<`¹⁰ liefert die Angaben zu einem Graphen, den die Software `>Mathematica<`¹¹ folgendermaßen darstellt (Abb. 2).

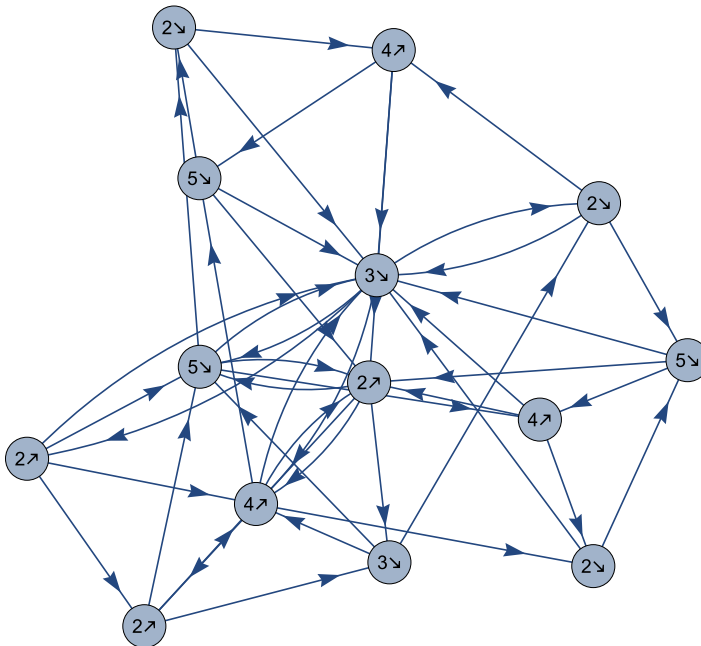


Abbildung 2: Prey-Graph für den dreistimmigen Oberquartkanon

¹⁰ Prey 2012, 104–105.

¹¹ <https://www.wolfram.com/mathematica>

Ebenso wie Tabelle 2 ist dieser Graph wohl zu komplex, um in der Praxis eingesetzt zu werden. Aber es zeigt sich, dass man die Fortschreitungsregeln sinnfälliger formulieren kann, wenn man sie ›händisch‹ herleitet. Als Ausgangspunkt nehmen wir die 22 Intervallpaare von Tabelle 1 (zweistimmiger Oberquartkanon). Damit der Dux mit dem zweiten Comes konsonant ist, muss die Summe zweier aufeinanderfolgender Intervalle ein Intervall ergeben, das mit einem Septimkanon kompatibel ist (Bsp. 4).

2. Comes

1. Comes

Dux

Quartkanon

Septimkanon

Beispiel 4: Konsonanzbedingungen für den dreistimmigen Oberquartkanon

Dadurch fallen drei von den 22 Intervallpaaren in Tabelle 1 weg, weil sie diese Bedingung nicht erfüllen, und zwar $2↗ 2↘$, $2↘ 2↗$ und $3↘ 3↘$. Für einen dreistimmigen Kanon muss man aber Folgen von drei Intervallen prüfen, denn es muss zusätzlich noch sichergestellt werden, dass es keine Parallelen zwischen Dux und zweitem Comes gibt. So ist zum Beispiel die Intervallfolge $2↗ 2↗ 2↗$ nicht möglich, da sie zu Quintparallelen führt (Bsp. 5).

$2↗$ $2↗$ $2↗$

Beispiel 5: Quintparallelen zwischen Dux und zweitem Comes

Insgesamt gibt es neun solcher Dreier-Intervallfolgen, die zu Parallelen zwischen Dux und zweitem Comes führen. Eine sinnfällige Gruppierung der Fortschreitungsregeln wäre die folgende (Tab. 3).

4↗ 2↘	2↗ 4↘
4↗ 2↘ (aber nicht 4↗ 2↘ 4↗)	2↘ 4↗ (aber nicht 2↘ 4↗ 2↘)
4↗ 3↘	3↘ 4↗
4↗ 5↘ (aber nicht 4↗ 5↘ 4↗)	5↘ 4↗ (aber nicht 5↘ 4↗ 5↘)
2↗ 3↘ (aber nicht 2↗ 3↘ 2↗)	3↘ 2↗ (aber nicht 3↘ 2↗ 3↘)
2↗ 2↗ (aber nicht 2↗ 2↗ 2↗)	
2↗ 5↘	5↘ 2↗
2↘ 3↘	3↘ 2↘
2↘ 5↘ (aber nicht 2↘ 5↘ 2↘)	5↘ 2↘ (aber nicht 5↘ 2↘ 5↘)
3↘ 5↘	5↘ 3↘

Tabelle 3: Fortschreitungsregeln für den dreistimmigen Oberquartkanon, sinnfällig gruppiert

Alle verbotenen Dreier-Intervallfolgen haben die Eigenschaft, dass man sich zwischen zwei Knoten hin- und herbewegt. (Dies ist auch einleuchtend, denn bei den verbotenen Dreierfolgen entstehen ja Parallelen zwischen Dux und zweitem Comes; somit muss das erste und das letzte Intervall dasselbe sein.) Wenn man gewissen Kanten (Verbindungslinien) im Prey-Graphen die zusätzliche Eigenschaft verleihen kann, dass sie nicht zweimal nacheinander in Gegenrichtung durchlaufen werden dürfen, lassen sich die Fortschreitungsregeln ziemlich übersichtlich darstellen. Man erhält dann einen ›kantengefärbten Prey-Graphen‹, bei dem man sich durchaus vorstellen kann, dass er auch für Improvisationssituationen praxistauglich sein kann (Abb. 3).

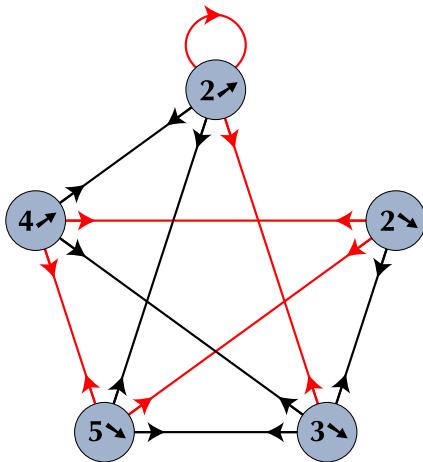


Abbildung 3: Kantengefärbter Prey-Graph für den dreistimmigen Oberquartkanon. Die roten Kanten dürfen nicht zweimal hintereinander (in Gegenrichtung) durchlaufen werden.

Der vierstimmige Oberquartkanon

Für den vierstimmigen Oberquartkanon liefert Preys Algorithmus »kgra« die Angaben zu einem Graphen, den die Software »Mathematica« folgendermaßen darstellt (Abb. 4).

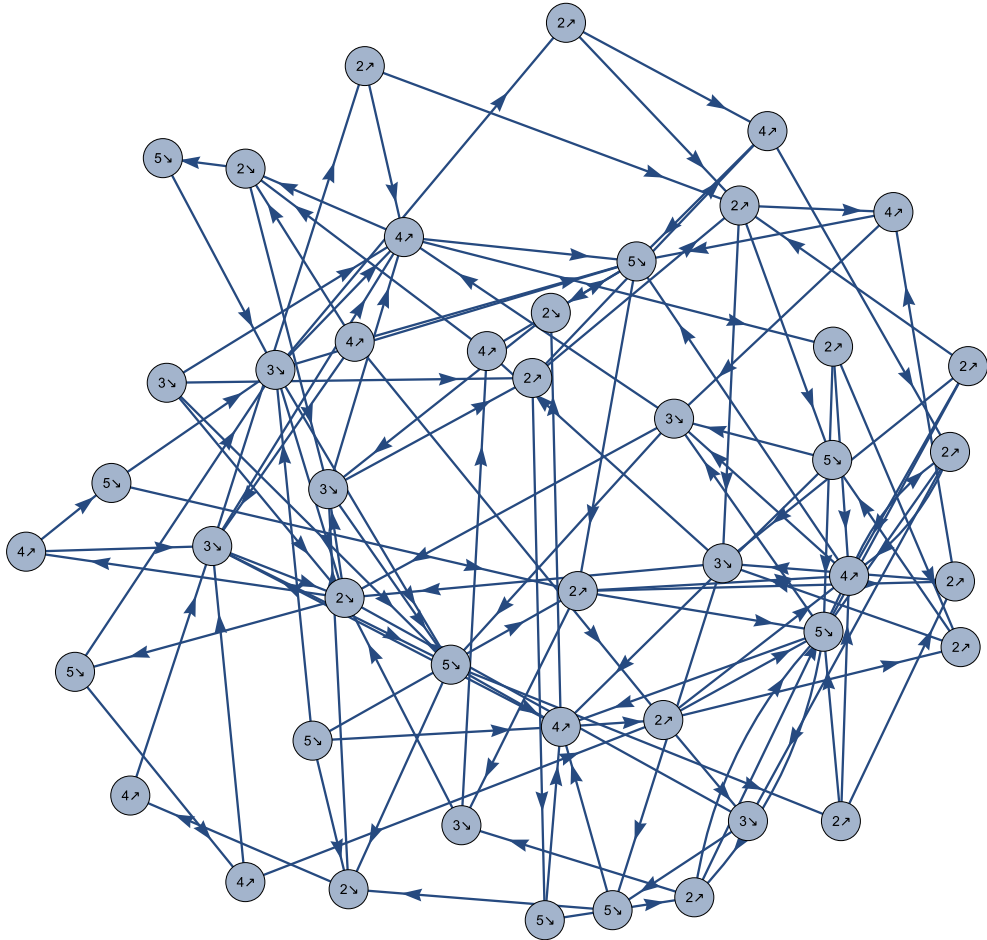


Abbildung 4: Prey-Graph für den vierstimmigen Oberquartkanon

Obwohl Mathematica versucht, die Darstellung des Graphen zu optimieren, ist das Resultat ziemlich abschreckend. Der LISP-Algorithmus des Autors generiert die folgende Liste von Vierer-Intervallfortschreitungen, die nicht minder furchteinflößend ist (Tab. 4).

4♯ 2♯ 4♯ 2♯	2♯ 4♯ 2♯ 4♯	2♯ 4♯ 3♯ 4♯	3♯ 4♯ 2♯ 4♯	5♯ 4♯ 2♯ 4♯
4♯ 2♯ 4♯ 3♯	2♯ 4♯ 2♯ 3♯	2♯ 4♯ 3♯ 2♯	3♯ 4♯ 2♯ 2♯	5♯ 4♯ 2♯ 2♯
4♯ 2♯ 4♯ 5♯	2♯ 4♯ 2♯ 5♯	2♯ 4♯ 3♯ 2♯	3♯ 4♯ 2♯ 5♯	5♯ 4♯ 2♯ 3♯
4♯ 2♯ 2♯ 3♯	2♯ 4♯ 3♯ 4♯	2♯ 4♯ 3♯ 5♯	3♯ 4♯ 2♯ 3♯	5♯ 4♯ 2♯ 5♯
4♯ 2♯ 2♯ 5♯	2♯ 4♯ 3♯ 2♯	2♯ 4♯ 5♯ 2♯	3♯ 4♯ 2♯ 5♯	5♯ 4♯ 2♯ 3♯
4♯ 2♯ 3♯ 2♯	2♯ 4♯ 3♯ 5♯	2♯ 4♯ 5♯ 3♯	3♯ 4♯ 5♯ 2♯	5♯ 4♯ 3♯ 4♯
4♯ 2♯ 3♯ 5♯	2♯ 4♯ 5♯ 2♯	2♯ 3♯ 4♯ 2♯	3♯ 4♯ 5♯ 2♯	5♯ 4♯ 3♯ 2♯
4♯ 2♯ 5♯ 4♯	2♯ 4♯ 5♯ 2♯	2♯ 3♯ 4♯ 2♯	3♯ 4♯ 5♯ 3♯	5♯ 4♯ 3♯ 2♯
4♯ 2♯ 5♯ 2♯	2♯ 4♯ 5♯ 3♯	2♯ 3♯ 4♯ 5♯	3♯ 2♯ 4♯ 2♯	5♯ 4♯ 3♯ 5♯
4♯ 2♯ 5♯ 3♯	2♯ 2♯ 4♯ 3♯	2♯ 3♯ 2♯ 4♯	3♯ 2♯ 4♯ 5♯	5♯ 2♯ 4♯ 2♯
4♯ 2♯ 3♯ 4♯	2♯ 2♯ 4♯ 5♯	2♯ 3♯ 2♯ 2♯	3♯ 2♯ 2♯ 4♯	5♯ 2♯ 4♯ 3♯
4♯ 2♯ 3♯ 2♯	2♯ 2♯ 3♯ 4♯	2♯ 3♯ 2♯ 5♯	3♯ 2♯ 2♯ 3♯	5♯ 2♯ 4♯ 5♯
4♯ 2♯ 3♯ 5♯	2♯ 2♯ 3♯ 2♯	2♯ 3♯ 5♯ 4♯	3♯ 2♯ 2♯ 5♯	5♯ 2♯ 2♯ 4♯
4♯ 2♯ 5♯ 3♯	2♯ 2♯ 3♯ 5♯	2♯ 3♯ 5♯ 2♯	3♯ 2♯ 5♯ 4♯	5♯ 2♯ 2♯ 3♯
4♯ 3♯ 4♯ 2♯	2♯ 2♯ 5♯ 4♯	2♯ 3♯ 5♯ 2♯	3♯ 2♯ 5♯ 2♯	5♯ 2♯ 3♯ 4♯
4♯ 3♯ 4♯ 2♯	2♯ 2♯ 5♯ 3♯	2♯ 5♯ 4♯ 2♯	3♯ 2♯ 4♯ 3♯	5♯ 2♯ 3♯ 2♯
4♯ 3♯ 4♯ 5♯	2♯ 3♯ 4♯ 2♯	2♯ 5♯ 4♯ 3♯	3♯ 2♯ 4♯ 5♯	5♯ 2♯ 5♯ 4♯
4♯ 3♯ 2♯ 2♯	2♯ 3♯ 4♯ 5♯	2♯ 5♯ 3♯ 4♯	3♯ 2♯ 3♯ 4♯	5♯ 2♯ 5♯ 2♯
4♯ 3♯ 2♯ 5♯	2♯ 3♯ 2♯ 4♯	2♯ 5♯ 3♯ 2♯	3♯ 2♯ 3♯ 2♯	5♯ 2♯ 5♯ 3♯
4♯ 3♯ 2♯ 4♯	2♯ 3♯ 2♯ 3♯	2♯ 5♯ 3♯ 2♯	3♯ 2♯ 3♯ 5♯	5♯ 2♯ 4♯ 3♯
4♯ 3♯ 2♯ 3♯	2♯ 3♯ 2♯ 5♯	2♯ 5♯ 3♯ 5♯	3♯ 2♯ 5♯ 4♯	5♯ 2♯ 3♯ 4♯
4♯ 3♯ 2♯ 5♯	2♯ 3♯ 5♯ 4♯		3♯ 2♯ 5♯ 3♯	5♯ 2♯ 3♯ 2♯
4♯ 3♯ 5♯ 4♯	2♯ 3♯ 5♯ 2♯		3♯ 5♯ 4♯ 2♯	5♯ 2♯ 3♯ 5♯
4♯ 3♯ 5♯ 2♯	2♯ 5♯ 4♯ 2♯		3♯ 5♯ 4♯ 2♯	5♯ 3♯ 4♯ 2♯
4♯ 3♯ 5♯ 2♯	2♯ 5♯ 4♯ 2♯		3♯ 5♯ 4♯ 3♯	5♯ 3♯ 4♯ 2♯
4♯ 5♯ 2♯ 4♯	2♯ 5♯ 4♯ 3♯		3♯ 5♯ 2♯ 4♯	5♯ 3♯ 4♯ 5♯
4♯ 5♯ 2♯ 2♯	2♯ 5♯ 2♯ 4♯		3♯ 5♯ 2♯ 2♯	5♯ 3♯ 2♯ 4♯
4♯ 5♯ 2♯ 3♯	2♯ 5♯ 2♯ 3♯		3♯ 5♯ 2♯ 5♯	5♯ 3♯ 2♯ 2♯
4♯ 5♯ 2♯ 5♯	2♯ 5♯ 2♯ 5♯		3♯ 5♯ 2♯ 4♯	5♯ 3♯ 2♯ 4♯
4♯ 5♯ 2♯ 3♯	2♯ 5♯ 3♯ 4♯		3♯ 5♯ 2♯ 3♯	5♯ 3♯ 2♯ 3♯
4♯ 5♯ 3♯ 4♯	2♯ 5♯ 3♯ 2♯			5♯ 3♯ 2♯ 5♯
4♯ 5♯ 3♯ 2♯	2♯ 5♯ 3♯ 5♯			5♯ 3♯ 5♯ 4♯
4♯ 5♯ 3♯ 2♯				5♯ 3♯ 5♯ 2♯
4♯ 5♯ 3♯ 5♯				5♯ 3♯ 5♯ 2♯

Tabelle 4: Intervallfortschreitungen für den vierstimmigen Oberquartkanon. Die von Mouton verwendeten Intervallkombinationen sind gelb unterlegt.

Interessant sind vor allem die Fortschreitungen, die *nicht* in Tabelle 4 stehen, aber im dreistimmigen Oberquartkanon (Tab. 3) möglich wären. Das sind zunächst einmal jene sieben Fortschreitungen, bei denen eine Dissonanz zwischen Dux und drittem Comes (Dezimkanon!) entsteht: 3♯ 5♯ 3♯, 3♯ 4♯ 3♯, 2♯ 5♯ 2♯, 2♯ 3♯ 2♯, 2♯ 4♯ 2♯, 2♯ 5♯ 2♯ und 2♯ 4♯ 2♯. Des weiteren gibt es nicht weniger als 24 Fortschreitungen (etwa 3♯ 5♯ 2♯ 3♯), bei denen Parallelen zwischen Dux und drittem

Comes entstehen. Naturgemäß ist bei diesen Fortschreitungen das erste und letzte Intervall dasselbe.

Bei dieser Situation nützt es nichts mehr, gefärbte Graphen mit speziellen Kanten einzuführen, die nur einmal durchlaufen werden können. Da eine Korrelation zwischen Intervallen besteht, die bis zu vier Noten auseinanderliegen, führt kein Weg daran vorbei, dass ein bestimmter Knotentyp mehrmals im Graphen vorkommt oder dass man den Graphen mit zusätzlichen Restriktionen versieht (siehe unten). Der Autor hat keine Methode gefunden, den Graphen von Abbildung 4 zu einer halbwegs übersichtlichen Darstellung zu vereinfachen. Umso interessanter ist es abzuklären, wieviele Kombinationsmöglichkeiten Mouton in seinem Kanon tatsächlich verwendet hat, anhand des folgenden Note-gegen-Note-Extrakts (Bsp. 6).¹²

Beispiel 6: Note-gegen-Note-Extrakt des Mouton-Kanons

Es stellt sich heraus, dass Mouton nur einen kleinen Bruchteil der möglichen Intervallfolgen verwendet, nämlich nur jene zwölf Folgen von 151, die in Tabelle 4 gelb unterlegt sind. Besonders prominent sind die ›Floskeln‹ $5\searrow 4\searrow 2\searrow$, $5\searrow 4\searrow 2\searrow$ und $3\searrow 4\searrow 2\searrow$ (die letzten beiden können beliebig oft wiederholt werden). Eine ganze aus diesen Floskeln bestehende Phrase $5\searrow 4\searrow 2\searrow 5\searrow 4\searrow 2\searrow 3\searrow 4\searrow 2\searrow 3\searrow$ wird gar wörtlich wiederholt. Die Fortschreitungen des Mouton-Kanons können auf folgenden einfachen Prey-Graphen reduziert werden, der zweifelsohne auch als Improvisationsvorlage dienen könnte (Abb. 5).

¹² Die letzten sechs Messuren wurden weggelassen, da sie sich nicht auf einen Note-gegen-Note-Satz auf Semibrevis-Ebene reduzieren lassen. Vgl. hierzu auch Notenbeispiel 5 bei Urquhart 2015.

Heikel ist die Eliminierung von Parallelen zwischen Dux und drittem Comes. Es handelt sich um Vier-Intervall-Kombinationen, bei denen das erste und das letzte Intervall identisch sind, mithin bewegt man sich im Graphen in einem Dreieck. Leider ist es nicht möglich, Dreiecksbewegungen pauschal zu verbieten, denn dann würde man in Mouton vorkommende Kombinationen wie $4\swarrow 2\swarrow 5\searrow 4\swarrow$ oder $2\searrow 3\searrow 4\swarrow 2\searrow$ auch eliminieren. Abbildung 7 zeigt, welche Dreiecksbewegungen nicht möglich sind.

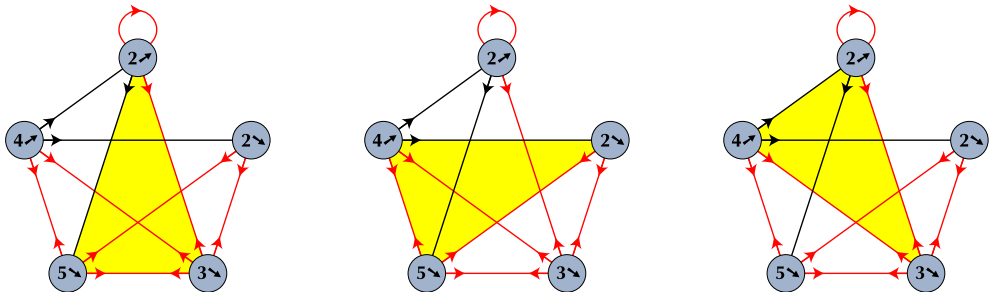


Abbildung 7: Verbotene Dreiecksbewegungen in Abbildung 6, die zu Parallelen zwischen Dux und drittem Comes führen

Die Frage, ob der Graph von Abbildung 6 mit den zusätzlichen Regularien von Abbildung 7 für eine Improvisation praxistauglich ist, lässt der Autor lieber offen.

Beispiele von algorithmisch generierten Oberquartkanons mit vier oder mehr Stimmen

Wie bereits erwähnt wurde, können die algorithmisch generierten Intervallfortschritungstabellen (wie Tabelle 2 oder Tabelle 4) in einen weiteren Algorithmus eingespeist werden, der daraus via Zufallsgenerator einen Kanon generiert. Vorher ist aber eine kleine Statistik interessant, nämlich wieviele Intervallkombinationen je nach Stimmenzahl möglich sind. Wie zu erwarten nimmt der Anteil der erlaubten Intervallkombination – verglichen mit den prinzipiell möglichen Kombinationen aus $5\searrow, 3\searrow, 2\searrow, 2\swarrow$ und $4\swarrow$ – mit steigender Stimmenzahl dramatisch ab (Tab. 5).

n : Anzahl der Stimmen	5^n : Anzahl der prinzipiell möglichen Kombinationen von n Intervallen aus $5\searrow, 3\searrow, 2\searrow, 2\swarrow$ und $4\swarrow$	Anzahl der mit dem n -stimmigen Oberquartkanon kompatiblen Intervallkombinationen	Prozentsatz
2	25	22	88 %
3	125	64	51 %
4	625	151	24 %
5	3.125	304	9,7 %
6	15.625	547	3,5 %
7	78.125	699	0,89 %
8	390.625	177	0,045 %

Tabelle 5: Anzahl der möglichen Intervallkombinationen je nach Stimmzahl

Um einen n -stimmigen Kanon mit Hilfe der entsprechenden Liste der erlaubten Intervallfolgen zu generieren, geht der Algorithmus folgendermaßen vor: Ein Zufallsgenerator wählt zunächst eine erlaubte Folge von n Intervallen aus. Die $n - 1$ letzten Intervalle aus dieser Folge werden dann abgeglichen mit den $n - 1$ ersten Intervallen aus der Menge der erlaubten Folgen. Von allen passenden Folgen wählt der Algorithmus diejenige aus, deren letztes, n -tes Intervall das kleinste ist. (Täte er das nicht, so erhielte man ästhetisch unbefriedigende, hilflos umherspringende Dux-Melodien.) Das Verfahren wird so lange wiederholt, bis die gewünschte Kanonlänge erreicht ist. Es kann – vor allem bei vielstimmigen Kanons – jedoch passieren, dass das Verfahren abgebrochen werden muss, weil keine passende Intervallfolge mehr vorliegt.

Vierstimmige Kanons generiert der Algorithmus erwartungsgemäß problemlos (Bsp. 7).



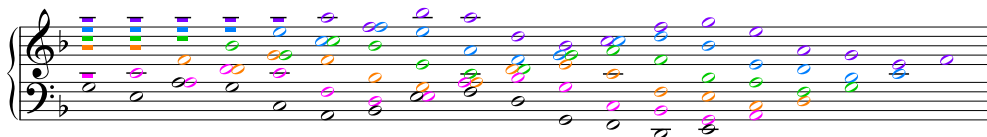
Beispiel 7: Algorithmisch generierter vierstimmiger Oberquartkanon

Bei fünfstimmigen Kanons tritt ein interessantes Phänomen auf: Der Algorithmus ›beißt sich fest‹ an einer Art Endlosschleife, aus der es offenbar keinen Ausweg gibt (Beispiel 8).



Beispiel 8: Algorithmisch generierter fünfstimmiger Oberquartkanon mit Endlosschleife

Interessanterweise ist das Problem der Endlosschleife bei sechsstimmigen Kanons weit weniger präsent – ein typisches Phänomen ist dort vor allem, dass der Dux die Tendenz hat, sich immer mehr nach unten zu schrauben (Bsp. 9).



Beispiel 9: Algorithmisch generierter sechsstimmiger Oberquartkanon

Zu guter Letzt seien noch Beispiele für sieben- und achtstimmige Kanons angeführt (Bsp. 10). Hier tritt vor allem das Phänomen auf, dass nach zehn bis 13 Notennoten keine Weiterführungsmöglichkeit mehr besteht und der Kanon abgebrochen werden muss.



Beispiel 10: Algorithmisch generierte sieben- und achtstimmige Oberquartkanons

Es gäbe noch viele Fragen, denen man nachgehen könnte: Könnte man für andere, komplizierte Kanontypen den Prey-Graphen computergestützt vereinfachen und praxistauglich machen? Welche ›Loops‹ gibt es bei welchem Kanontyp? Wie lange ist der längstmögliche sieben- oder achtstimmige Kanon? Was passiert, wenn man weniger restriktive Arbeitshypothesen verwendet als diejenigen in diesem Aufsatz?

Wie man oben sieht, gibt es beim Oberquartkanon ab der Fünfstimmigkeit Probleme. Abschließend sei deshalb die Bemerkung erlaubt, dass Mouton bei dem vierstimmigen Oberquartkanon offenbar eine Grenze erreicht hat, die nur schwer überschritten werden kann.

Literatur

- Berger, Karol (1987), *Musica ficta: Theories of Accidental Inflections in Vocal Polyphony from Marchetto da Padova to Gioseffo Zarlino*, Cambridge: Cambridge University Press.
- Burn, David (2001), »Further Observations on Stacked Canon and Renaissance Compositional Procedure: Gascongne's ›Ista est speciosa‹ and Forestier's ›Missa L'homme armé‹«, *Journal of Music Theory* 45/1, 73–118. <https://doi.org/10.2307/3090649>
- Gauldin, Robert (1996), »The Composition of Late Renaissance Stretto Canons«, *Theory and Practice* 21, 29–54.
- Gosman, Alan (1997), »Stacked Canon and Renaissance Compositional Procedure«, *Journal of Music Theory* 41/2, 289–317.
- Janin, Barnabé (2012), *Chanter sur le livre*, Langres: Éditions Dominique Guéniot.
- MacCracken, Thomas G. / Mary Beth Winn (2014), *Joannes Mouton – Opera omnia*, Corpus Mensurabilis Musicae (CMM) 43, Bd. 5: *Missa Sine nomine I & II, Credo a 4, Magnificat et Cantiones*, American Institute of Musicology.
- Ott, Immanuel (2014), *Methoden der Kanonkomposition bei Josquin Des Prez und seinen Zeitgenossen*, Hildesheim: Olms.
- Prey, Stefan (2012), *Algorithmen zur Satztechnik und ihre Anwendung auf die Analyse*, Dissertation, Universität Osnabrück. <https://repositorium.ub.uni-osnabrueck.de/handle/urn:nbn:de:gbv:700-2012102410434>
- Urquhart, Peter (1997), »Calculated to Please the Ear: Ockeghem's Canonic Legacy«, *Tijdschrift van de Koninklijke Vereniging voor Nederlandse Muziekgeschiedenis* 47/1–2, 72–98.
- Urquhart, Peter (2015), »Remarks on Some Chansons by Jean Mouton, and Related Matters«, *Tijdschrift van de Koninklijke Vereniging voor Nederlandse Muziekgeschiedenis* 65/1–2, 37–65.

Pierre Funck

© 2021 Pierre Funck (pierre.funck@mac.com)

Zürcher Hochschule der Künste

Funck, Pierre (2021), »Die Mouton-Maschine. Ein algorithmischer Blick auf stacked canons in der Oberquart«, in: *Populäre Musik und ihre Theorien. Begegnungen – Perspektivwechsel – Transfers. 17. Jahreskongress der Gesellschaft für Musiktheorie und 27. Arbeitstagung der Gesellschaft für Populärmusikforschung Graz 2017* (= GMTH Proceedings 2017), hg. von Christian Utz, 67–84.
<https://doi.org/10.31751/p.42>

eingereicht / submitted: 04/11/2018

angenommen / accepted: 23/11/2018

veröffentlicht / first published: 05/01/2021

zuletzt geändert / last updated: 05/01/2021