Melodic Swarms: using swarm algorithms for the generation of melodies in a harmonic context

Sam van Tienhoven

Graduation Thesis

Media Technology MSc program
Leiden University
January 2018

Thesis Advisors: Edwin van der Heide, Daniel Bisig, Philippe Kocher

Abstract

Artificial intelligence techniques have been applied to music composition in various ways, among which are swarm algorithms. The self-organizing property of swarm algorithms is similar to certain musical behavior. Research on applying swarm algorithms to composition has led to systems being able to create complex soundscapes and musical textures. However, the composition of melodies using swarm algorithms has not been extensively studied. Using an experimental approach, we have created multiple swarm algorithm systems that generate melodic phrases based on a given chord progression. These approaches are then assessed both on the basis of their individual merits and in comparison to each other. Concretely, we try to distill the specific traits, benefits and disadvantages of using swarm algorithms for the generation of melodies. We then discuss further research and possible applications of melodic swarms.

1 Introduction

1.1 Artificial Intelligence and Music

Since the advent of artificial intelligence (AI) research, applications have been found in art and music. Art and music is an extraordinary interesting subject for AI research as creativity is a fundamental aspect of intelligence. To design intelligent systems one must design creative systems. Art and music is a very common representation of creativity, thus making it an interesting subject for AI research.

Traditionally, computers are not very good at being creative. A computer follows strict rules set by a programmer and handles input strictly according to these rules. Art is more than applying learned rules, it implies concepts such as creativity, originality and expression. Art is an emergent property of the human mind which can activate emotion and provoke thought. For a computer to create art we must find a way to develop algorithms with these properties. Art and music do have underlying frameworks, be it perspective and color for painting or rhythm and harmony for music. These frameworks are a good starting point for creative computer systems. Creative frameworks can be formalized into mathematical models which can be processed by a computer. However, these rules alone do not suffice for the creation of art. Some form of AI must process these rules in a way that can create something novel and meaningful.

A range of AI techniques has been applied to music for various purposes. These purposes range from harmonization to rhythm generation and the writing of lyrics. One way to distinguish between different AI techniques applied to music is by dividing them between analytical and generative approaches. Analytical techniques such as machine learning and probabilistic approaches aim to extract patterns from existing data. In a way, analytical systems are able to learn musical behavior, instead of using programmed musical knowledge described in code. For instance, FlowMachines analyzed the entire corpus of Beatles songs in order to compose a new song in the style of the Beatles [1]. Analytical techniques excel in composing structured music without requiring an extensive theoretical framework.

In contrast to analytical approaches stand generative approaches. These systems generate

data according to some process and translate this to musical output using a theoretical framework. The method of generation varies from randomly generated numbers to stochastic processes and biologically influenced algorithms such as swarm algorithms. The emergent nature of some of these algorithms is reflected in the resulting musical output. The mapping between the generated data and the musical output is an important factor and has a large influence on these emergent characteristics.

1.2 Research question

Generative approaches, and specifically swarm algorithms, are typically applied in systems that create interactive sonic soundscapes with textural depth or systems that are able to improvise in a free form style. In contrast, analytical systems have been successful in composing melodies and chord progressions in a more conventional idiom such as western pop music. Then the question arises: how can the emergent and unpredictable qualities of generative systems be harnessed in order to compose structured melodies? This question is interesting because it asks if a fundamentally non-musical system (in contrast to analytical systems, which need to be trained with existing musical input) can create musical structures.

In this paper we will specifically focus on melody generation with swarm algorithms in a given harmonic context. Swarm algorithms are algorithms inspired by biological phenomena such as flocks of birds, schools of fish and colonies of ants. Their behavior is simulated by letting a multitude of digital individuals, called agents, interact with each other according to a set of basic rules. In essence, we want to create a meaningful and effective translation from swarm behavior to melody.

To find a translation we will take an experimental approach and create multiple systems that generate melodies with swarm algorithms. One important aspect is that these systems will all use harmonic context. This means a specific chord progression is used to offer a context for the melodic output, while also influencing the translation and swarm behavior. Making a translation to music means the swarm must be subjected to a theoretical framework. The balance between inherent qualities of the swarm and a theoretical framework then becomes an important factor. How much of the melody is the result of the theoretical framework? Can we distill the specific musical traits of the swarm algorithm?

And what exactly is the added value of the swarm as opposed to generating data differently?

1.3 Melody

For clarity, we want to define what we mean with structured melodies. A melody is a sequence of pitches in time, which is often repeated and forms the most recognizable part of a piece of music. It works in unison with harmony, which is the combination of pitches in order to form chords. The time aspect of a melody is determined by rhythm, which is the timing of the notes. For our research we have chosen to focus on tonal music with a tonal center or a key around which it revolves.

The type of structured melodies we wish to generate typically contain concrete values from a single musical scale. A piece of music often has a tonal center, or a key, around which it resolves. Both the melody and the harmony are based on this key. For instance, if a piece is in the key of C the notes in a melody will typically consist exclusively of C, D, E, F, G, A or B notes. Likewise, the harmony will consist of C major, D minor, E minor, F major, G major, A minor or B diminished chords. In tonal music there is often a strong connection between the melody and the harmony, where notes from the underlying harmony have a strong effect when also used in the melody.

The melodies we wish to generate follow commonly used rules in composing melodies. For example, maintaining a balance between chord tones and non-chord tones to keep a melody stable but interesting. It is also common that intervals between notes are kept small. Larger intervals can be used for effect but too many large intervals will lead to an incoherent melody. Also, the tension and release within a melody is always dependent of the harmonic context. The timing of the notes in a melody is as important as the pitch of the notes and functions similarly. Notes in a melody can vary between shorter and longer. However, there is often a rhythmic center, or a pulse, in melodies around which these variations revolve.

The melodies we wish to generate are inspired by melodies present in popular music. We want to generate melodies that follow the stylistic properties of vocal melodies present in pop, rock, folk, blues and other conventional western styles. This stands in contrast with

typically instrumental melodies that can be found in for instance free-jazz, avant-garde, free-form improvisation or some types of exotic music.

1.4 Swarm Algorithms

Swarm algorithms are algorithms inspired by biological phenomena such as the forming of ant colonies and the flocking of birds. Swarm algorithms typically consist of a number of individuals, called agents, which interact with each other in a defined space according to simple rules. Individuals can also react to other factors within their environment or external input. One of the main characteristics of swarm algorithms is that they are self-organizing. Structure arises solely from interactions between the individuals, without any centralized control.

The archetypical swarm algorithm is Boids, a program developed by Craig Reynolds in 1986 [2]. Boids, short for bird-oid object, is a program that simulates the flocking of birds. The program introduced at least three rules of interaction that have since become a staple in swarm algorithms: separation, alignment and cohesion. Separation makes individuals avoid collision with other individuals in the swarm. Alignment makes individuals follow the average direction of other individuals. Cohesion makes individuals move towards average position of the swarm. The contradictory nature of these rules leads to a feedback loop which creates emergent behavior. Typically, the strength of these rules can be adjusted and additional rules can be added to expand the functionality or increase the complexity.

Swarm algorithms have typically been applied to interactive free-form improvisation or soundscape generation systems since this matches the self-organizing property of swarm algorithms especially well. However, we argue that a strong analogy can also be made between the underlying principles of swarm algorithms and melody generation. Separation ensures individuals in a swarm will not be too close to each other. The same goes for notes in a melody. A melody becomes boring if a note is repeated many times so there must be a force driving notes apart from each other to create interesting melodic patterns. However, melodies should not become too erratic and use too many too large intervals. Cohesion helps notes to stick together and create unity and rest within a melodic movement. Alignment ensures that all notes will move in a similar direction. This results in successive melodic phrases maintaining some similarity.

1.5 Mapping strategies

As mentioned earlier, the translation of the output of a swarm algorithm to music is one of the most important aspects of a musical swarm algorithm system. We described this translation by using the term theoretical framework. A theoretical framework is a set of rules based on music theory which turns numerical output into musical output. We now will discuss some possible mapping strategies for translating a swarm to music [3]. These mapping strategies are not mutually exclusive.

The simplest form of translating swarm output to music is *direct parameter mapping*. This mapping consists of forming a direct link between a parameter of a swarming agent to a musical parameter. For instance, the position of an agent on the y-axis in a two-dimensional swarming space could be linked to the pitch of a note. This results in an intuitive relationship between the visual representation of the swarm and the resulting melody.

Proximity-based mapping is similar to direct parameter mapping but instead of the position it uses the distance between agents. This means you can exchange the intuitively logical but conceptually often arbitrary relationship between for instance y-axis position and pitch for a more intrinsically meaningful relationship. The orientation of a swarm will always be arbitrary unless for instance you introduce some representation of gravity. For this reason using relative position seems more meaningful than using absolute position. Variations on proximity-based mapping could be relative speed, or deviation from the average position of the swarm.

One-to-one mapping and *many-to-one mapping* describe the relationship between the number of agents and the number of notes. One-to-one mapping means that each agent will produce a note, many-to-one mapping means that multiple agents will produce one note. This second approach produces one note which is the result of the swarm as a whole and can be used effectively when the swarm is required to produce a melody which reflects all individuals in real time.

Especially interesting is the term *conceptual mapping*. Conceptual mapping means that the relationship between the algorithm and its musical output is artistically meaningful, so that the underlying algorithm becomes an integral part of the artistic idea. An example of this is AtomSwarm (see 1.6). Here an expanded biological swarm simulation is conceptually linked to its output by making a complex analogy between its behavior and music.

1.6 Relevant Work

Swarm Music is an interactive system created in 2001 by Tim Blackwell [4]. The system is able to generate melody, harmony and rhythmic patterns with swarm algorithms and generates free-form improvisation. Swarm Music is an adaptation of the Boids algorithm with expanded functionality. The core of Swarm Music consists of five individuals that move around in a 7-dimensional space with a certain speed and direction [5]. Each of the seven dimensions represents a musical quality, some on a note-level such as loudness, duration and pitch while other dimensions represent higher level qualities such as similarity to other notes in the swarm and onset time between notes. He primarily uses direct parameter mapping and proximity mapping although there is also a conceptual link.

The position of the individuals of the swarm is updated with a specific time interval. Each time the position of the swarm is updated, the entire set of individuals is interpreted as a set of musical notes. This set is then played in a certain order. For instance, the set of notes can be played in ascending order of pitch or can be played from soft to loud. Meta-level dimensions are also interpreted. For instance the position of notes in the onset-time dimension could be interpreted as: play the first two notes of the set simultaneously, play a single note and then play the last two notes of the set simultaneously again. To make the output sound more musical, some variables can be controlled by introducing restrictions. These restrictions can be very global, such as limit the possible pitches to notes appearing in the C major scale or to notes below the middle C. A restriction could also be that the notes have to be played in the average pulse of the set, instead of their individual pulses. An interesting proposal by Blackwell is to evaluate the musical output by using a musical Turing test. Experienced subjects will be exposed to both swarm- and human-generated melodies and will have to guess what generated the melody.

AtomSwarm is a swarm algorithm based music system first published by Daniel Jones in 2008 [6]. AtomSwarm is used to create atonal textural pieces. Just like Swarm Music, AtomSwarm is based on Reynold's Boids algorithm. AtomSwarm introduces a range of biologically inspired concepts to a swarm algorithm music system not found in Swarm Music including genes, hormones, food, reproduction, aging, death, and day and night cycles. Each individual has a genome which encodes certain properties such as their attraction to other individuals and their rate of aging. Individuals also have hormone levels of several different hormones. These include melatonin, which makes them move slower, and testosterone which increases their likelihood of reproducing. Hormone levels are affected by the environment of the individuals, if it is night more melatonin is produced and testosterone increases with aging and decreases after reproduction. The environment also contains food, which individuals need to consume to stay alive and viruses, which can alter their genes. Individuals can reproduce with each other creating offspring with a combination of the genes of the parents. Finally, individuals can die of aging or starvation.

All these factors create a very complex and dynamic swarm with a varying population which has to be translated to music. Unlike Swarm Music, the musical output of AtomSwarm is not determined by the position of the individuals. Instead the genes of an individual encode when an individual is sonically activated. This can be a certain speed threshold or collision with another individual. Genes also control what aspect of an individual is translated to sound and in which way. For instance, the speed of an individual is translated into pitch, or in more complex cases, the relative speed of an individual is translated to the frequency of an oscillator. This means that AtomSwarm uses a complex conceptual mapping combined with advanced forms of proximity mapping.

Tatsuo Unemi and Daniel Bisig created a system which is able to interactively generate musical output [7]. The system consists of a swarm consisting of 64 individuals moving in a 3-dimensional space which can be controlled by human gestures, analogous to a conductor influencing an orchestra. Each individual has an x, y and z position and can also be playing varying instruments. The x-axis corresponds to stereo panning, the y-axis corresponds to pitch and the z-axis to loudness. This creates a very intuitive translation of the visual state of the swarm to the audio output. A human can steer the swarm using gestures. In advance a human can also define the chord progression, scale and selection order of the swarm. For

instance, the swarm can be instructed to rotate through individuals in a fixed order or a random order. These human instructions create restrictions to the swarm which leads to more musical results.

SoundSwarm is a system created by Tavares and Godoy which is generates ambient electronic music [8]. SoundSwarm functions by letting a swarm algorithm search for optima using test functions. It is especially interesting because it uses a purely conceptual mapping. They made an analogy between six aspects of the swarm such as the speed, spread and alignment and key concepts such as tension, serenity and simplicity. These concepts were then in turn related to musical aspects such as echo, loudness and duration. This approach leads to a very strong conceptual relation between intrinsic aspects of the swarm and musical output.

Melody Generator is a system developed by Dirk-Jan Povel in 2010 [9]. It is a system primarily focused on generating melodies given a harmonic context. Melody Generator generates melodies using random numbers subjected to a rigid theoretical framework. In this case, it's the theoretical framework which is most interesting to us. Specifically, it has interesting ways of handling pitch and tempo. Rhythm is quantified by adding weights to various positions. The first beat of a measure has the highest weight. Depending on what time signature the piece is in other beats are assigned other weights. These weights are eventually used to calculate the metrical stability of a measure. Strong notes are higher weight beats such as the first beat, and then other notes are divided over the other beats. Similar restrictions are applied to pitch. Non-chord tones are used only if they are resolved to chord tones quickly after.

2 Methods

2.1 Experimental approaches

We will create multiple experiments. In all our experiments we will research a different method of generating a melody on top of a given chord progression with swarm algorithms. All approaches have a different mapping of the swarm to music. The three general approaches are: direct interpretation, target areas and prescience. Direct interpretation will directly translate the individuals of a swarm to musical notes using a combination of direct

parameter mapping and proximity-based mapping. Target areas will focus on enhancing the space in which the swarm moves with target areas which trigger events. Prescience will use prior knowledge of the chord progression to adapt the interpretation of the swarm.

The goal of these experiments is to create systems we can assess in regard to our research question. Each system will have a different level of control over the swarm, varying restrictions and differences in mapping. The choices we make for our restrictions and mapping will be elaborated on in the discussion. The melodic output of our systems will be used for comparison between approaches and to other systems.

2.2 Basic implementation

Before we discuss our experimental approaches we will first define a rudimentary system which will be the basis for experimentation. The core of this system is an implementation of a swarm algorithm based on the Boids algorithm. We use a basic implementation created by Daniel Jones which he used for AtomSwarm. This implementation consists of a 2-dimensional space in which a variable number of individuals can move around. These individuals interact with each other according to three rules: cohesion, alignment and separation. The parameters of these rules can be adjusted.

The output of all systems will be midi signals. The advantage of using midi is that it is easy to work with and offers a lot of flexibility for the sound we wish to produce. All our approaches will be created using Processing, using the MidiBus library for handling midi output. A melody typically has a beginning and an ending instead of being a constant stream of notes. The output of our approaches must resemble this by delivering concrete melodic phrases[1] where a melodic phrase is defined as a distinct succession of notes. The way we will achieve this differs per approach and will be discussed in the relevant subchapter.

For the sake of simplicity all systems will be restricted to the C major scale only. This means that only natural notes (no sharps or flats) are produced and the harmony will

[1] A melodic phrase differs from a melody. A phrase is a group of notes which express a melodic idea. A phrase can be seen as the smallest subset of a melody. It's analogous to grammar where a melodic phrase is a group of words that make sense, but is not a complete sentence per se.

always consist of C major diatonic chords. In addition, the harmony is restricted to containing only triads. Note lengths are quantized to commonly used note lengths such as whole, half and quarter notes. We provide a more detailed discussion of the advantages and limitations of this approach in the discussion.

2.3 Approach 1: Direct interpretation

Our first experimental approach will focus on making a direct and detailed translation from the swarm to melody using a combination of direct parameter mapping and proximity-based mapping. We create a relationship between the harmony and swarm by using digital pheromones which influence swarm behavior. The mapping is inspired by SwarmMusic, as is the use of pheromones. However, the approach introduces a novel way of using pheromones to influence the swarm and uses alternative mapping relationships.

Firstly, we use a discrete interpretation. This means a snapshot of the swarm is taken with a specific time interval. The position of the individuals in the snapshot is then translated to musical notes. We use a one-to-one mapping where each individual becomes a note. We define each snapshot to constitute to a melodic phrase. Because a melodic phrase does not have to consist of an equal number of notes we randomly add or remove individuals of the swarm to create a varying number of notes per phrase. The position of all the individuals is calculated with a variable update rate. Each individual has three variables: pitch, velocity and duration.

Pitch	*Higher position on y-axis → higher pitch*
Velocity	*Closer to center of swarm → louder*
Duration	*Higher speed → shorter duration*
Update rate	*Higher average speed → higher update rate*

Table 1: interpretation of swarm for approach 1

The pitch of a note is related to its position on the y-axis, where a higher position leads to a higher note. This is based on musical notation where a higher position on a musical staff indicates a higher note. The pitch of a note is restricted to notes of the C major scale with a range of 8 octaves. Each pixel on the y-axis corresponds to a note of the C major scale.

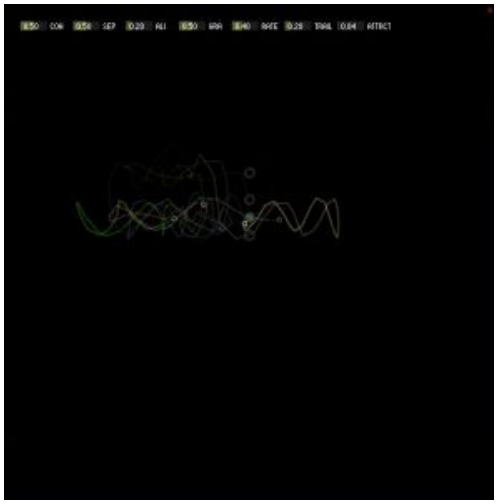
The velocity[2] of a note is related to the proximity of an agent to the center of mass of the swarm, where being closer to the center of mass increases the velocity. Being close to the center of the swarm means that an individual represents the average of the swarm and that the average distance to other individuals is relatively small. This average is represented by notes being played with more velocity. It also means that a note will be louder when the interval with other notes is smaller. We experiment with introducing a threshold which transforms notes with a velocity below a certain level to a rest.

The duration of a note is related to the speed of an agent, where a faster agent produces a shorter note. If a note is moving fast it means that it is very active and its pitch and volume values are changing rapidly. Because an individual will reflect a certain pitch and volume value for a short period of time its duration will also be a short period of time. Contrastingly, agents that have a relatively stable position will produce longer notes.

The update rate of interpretation of the swarm is related to the average speed of the swarm, where a higher average speed means a higher update rate. Similarly to the individual notes, if the swarm is moving rapidly, agents are very active and their relevant musical parameters change quickly. This excited state will be reflected in the swarm going through more iterations between each round of translation. It is important to note that this does not lead to a positive feedback loop. The update rate simply constitutes the number of iterations the swarm has between each snapshot. This allows the tempo of the output to remain constant.

[2] It is important to note that velocity and volume are two distinct concepts. Velocity refers to the strength with which a note is played. If you strike the crash of a drum kit softer, it will not only be lower in volume but also produce a different sound. The same goes for piano keys to a certain extent.

Low attraction



High attraction

To allow the swarm to adapt to a harmony we introduce pheromones which influence the swarm. The notes of a triad from the harmony will become four pheromones placed on a position on the y-axis corresponding to the pitch of the tonic, third, fifth and octave of the provided chord. The individuals of the swarm will be attracted to one of these pheromones. The pheromone a swarm is attracted to is chosen randomly to create varied results for each round. The strength of attraction can be adjusted in real time in order to experiment. Stronger attraction to pheromones means that duplicate notes will be present relatively often. Stronger attraction also leads to a higher presence of chord tones.

The harmony will be a preset iteration of four chords of equal duration and with a constant tempo. We define each chord to last one bar. A snapshot of the swarm is made and translated at the start of each bar. If a melodic phrase lasts longer than one bar, and thus overlaps with the next chord, no new snapshot will be made until the melody is finished. This is to prevent overlapping of melodies while also introducing occasional space between melodic phrases.

In order to achieve more structured melodic phrases we control the order in which the generated notes are played. The musical qualities of a melodic phrase arise from the properties and order the notes in it. For instance, a phrase feels more resolved if it ends on a chord tone. Vica versa, ending a phrase on a non-chord tone creates tension. A phrase can

have an upward or downward motion or can be more chaotic with bigger intervals between subsequent notes.

Depending on what notes are in a snapshot, notes are rearranged in a specific way. We will define three possible types of melodic sequences: ascending, descending and hyperbolic phrases. We will enforce the creation of these melodic phrases according to a number of rules which are applied in order. Rules are skipped when the notes are not available.

- A phrase must always start with the active pheromone.
- A phrase must always start and end with a chord tone.

- If *ascending*, a phrase must always end with the highest chord tone.
- If *ascending*, the remaining notes are sequenced in ascending order.

- If *descending*, a phrase must always end with the lowest chord tone.
- If *descending*, the remaining notes are sequenced in descending order.

- If *hyperbolic*, a phrase must always end with the same tone as which it started with.
- If *hyperbolic,* the highest or lowest note must be in the middle of the phrase.

To determine which type of sequencing is applied the raw notes produced by the swarm are first analyzed. When the number of unique notes is larger than half of the total notes we choose between ascending and descending. A large number of unique notes means it is possible to create a defined ascending or descending melodic phrase. If the number of unique notes is smaller than half of the total notes we choose hyperbolic sequencing. A small number of unique notes means that there is a relatively large amount of duplicate notes. This means there is relatively little movement in the melody and that a hyperbolic sequence can be effectively created. There is always a random chance notes will be placed in a random order. This functions partly as control group, and partly to see if interesting phrases are also produced without sequencing. This chance is equal to the chance of an ascending, descending or hyperbolic phrase being generated.

2.4 Approach 2: Target areas

Our second approach is a less direct translation of the swarm to music. Instead the general approach is to embed the 2-dimensional swarm space with target zones which trigger events. The swarm will not produce any sound unless a sound is triggered by an agent entering a target zone. The use of target areas introduces a range of extra variables which enhance the control over the output without affecting swarm behavior.

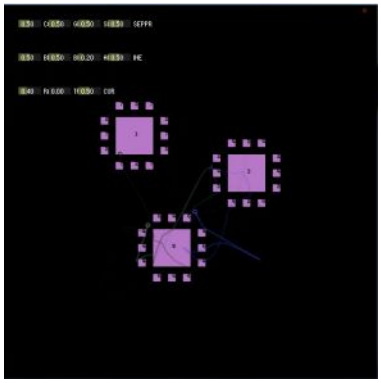
Target areas will be divided over the 2-dimensional space in a preset pattern. We experiment with three variations. The pitch of a note is determined by the pitch variable of a target area. Velocity of the produced note is determined by the speed with which an agent enters a target area. The analogy is that a target area is a bell and an agent is a solid object. When an object hits a bell faster, the produced note will be louder. The effect that an excited agent will produce a louder note is interesting to experiment with and also adds diversity to our mappings in addition to our direct interpretation approach.

The duration of a note is determined by the time an agent spends inside a target area. Essentially, this is similar to the direct interpretation approach since a slow note will produce longer notes than a fast note. The only difference is that this effect is amplified for chord tones since chord tones are represented by larger target areas in all variations. Instead of using the snapshot method of the direct interpretation approach we use a continuous interpretation for target areas. This means that the swarm is sonified in real time. This poses the challenge of dividing the output in melodic phrases, since the system is able to produce a constant stream of notes. The way we approach this is by enforcing a maximum number of notes that can be played per chord. This maximum number is randomly chosen within a range of 8 to 12 notes.

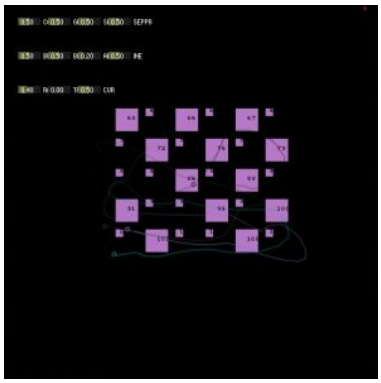
The **first variation** is a strong chord tone-based pattern. The tonic, third and fifth will be represented by large target areas. These large areas will be surrounded by multiple small areas representing nearby non-chord tones. For instance, the third will be surrounded by small areas representing the second and the fourth steps of the major scale. The intended effect is that chord tones will be played relatively often and non-chord tones will be played either before or after a nearby chord tone. There is a gap between the non-chord tone target areas which allows agents to play chord tones without playing non-chord tones.

The target areas are able to react to underlying harmony by changing the pitch value of a target area. If the harmony is a C major chord the large target areas will represent the C, E and G notes and the small areas will represent the D, F, A and B notes. Now if the chord changes to a D minor the position of the areas will remain the same but their corresponding notes will change to D, F and A and E, G, B, and C respectively.

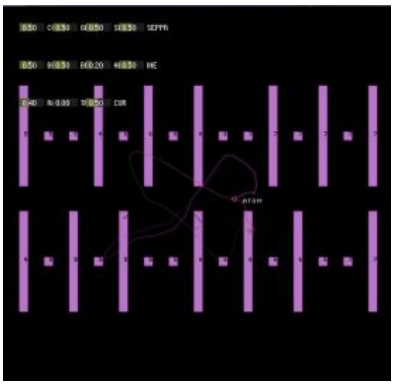
Because the order of the notes is dependent on the movement of the individuals, we use a many-to-one mapping. This means the average position of the swarm is used as an active agent. If we would use a one-to-one mapping there would be no coherence between the notes and the pattern would be ineffective.



chord based



square distribution



rectangular distribution

The **second variation** is distribution where target areas are placed semi-randomly within the swarm space. Target areas can have pitch values from all notes of the C major scale and are roughly grouped by pitch value. It reacts to the harmony by enlarging areas that represent chord tones. When the harmony changes the new chord tones will become large and all non-chord tones will become small. This means that the pitch value of a target area will remain constant and the size becomes variable. This approach requires a smaller theoretical framework and instead uses a more probability-based technique. This approach can use a one-to-one mapping if all the pitch values of the target areas are confined to a relatively small range, for instance 2 octaves.

Our **third variation** attempts to strike a balance between the first two variations. It uses rectangular target areas which are arranged in two rows of two octaves. The upper and lower row are roughly the same notes but shifted a perfect fifth. Again, chord tones will be enlarged and non-chord tones will be small. For this variation we experiment with either using a many-to-one mapping or using a one-to-one mapping with a small number of individuals.

The effect of this mapping is that when an agent moves in the horizontal direction it will have a large chance of playing a nearby chord tone and a smaller chance of playing a nearby non-chord tone. When an agent moves in a vertical direction it will have a large chance of playing a fifth interval. The fifth interval is a very stable and harmonious interval, which shares a similar quality to the octave.

2.5 Approach 3: Prescience

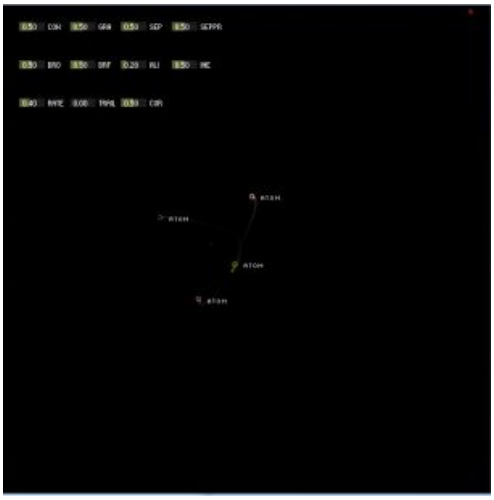
Our third and final approach is based on implementing some form of prior knowledge of the chord progression. It is inspired by human musicians who are typically conscious of the full chord progression while improvising a melody over it. This knowledge is used to adapt a melody to upcoming chords by for instancing playing chord tones from the next chord before that chord is being played. This creates a strong cohesion between melody and harmony and also creates more tension and release within a melody.

For this approach we combine the rigid interpretation used in the first approach with free swarm behavior used in the second approach. Again, agents move through a two-dimensional space. The y-axis represents pitch, but in this approach it only consists of chord tones. Velocity and duration is calculated identical to approach 1.

The system iterates through a predefined chord progression consisting of four chords from the C major key. The duration of these chords is stable and identical for all chords. The transition between two chords generates a special transitory state of the system. During this state, pitch values can not only be chord tones from the current chord but also chord tones from the upcoming or previous chords. This is illustrated in the image

to the right. The red zone is notes from the C major chord only, the yellow zone is notes from the D minor chord only. In the orange zone chord tones from both chords can be picked.

Identical to the target areas approach, a continuous interpretation is used. Experiments will be done with many-to-one mapping and one-to-one mapping. It is expected that many-to-one mapping will lead to a more cohesive melody while one-to-one mapping offers a more conceptual mapping similar to approach one. The problem of using many-to-one mapping is that velocity and duration will become uniform if we use the mapping from approach 1.



3 Results

3.1 Examples

We implemented all approaches and variations mentioned above and subjected these to identical chord progressions. We generated 12 examples which can be retrieved from here[3]. The approaches and settings used to generate each example can be found in Appendix 1.

We will examine and compare the examples within one approach. We will specifically discuss the general structure, variability and predictability of the generated melodies. The melodies will be interpreted on basis of the techniques used within one approach and the settings used to generate the example. Also, comparisons between other (sub)-approaches will be made.

[3] https://goo.gl/CtYPgE

3.2 Approach 1

Our first approach is able to create resolved and stable melodies. The melodies generated with the sequencer active are typically more resolved than the melodies without sequencing since they start and end on a chord tone. If the melodies would all start and end on the same chord tone they would become monotonous. Using a randomly chosen chord tone to start and end the phrase with keeps the melodies more unpredictable. It also increases the tension within the melodic phrases since ending on the third or the fifth has a less resolved feeling than ending on the first. Phrases that are not sequenced can provide interesting results but can become chaotic if the distance between swarm individuals becomes larger.

The rhythmical patterns that arise are not predictable since they are not enforced by a sequencer. All rhythmic patterns that arise are direct reflections of the behavior of the swarm. In *example 1* at the 17 second mark, a melodic phrase appears where the notes simply go up and back down three steps of the C major scale starting at C, enforced by the hyperbolic sequencer. The rhythmic pattern however is not enforced yet sounds structured without being predictable. The phrase that directly follows at the 21 second mark has a similar melodic structure but a slightly different rhythmic pattern. This is an example of an interesting sequence of two phrases. The phrase that follows at the 25 second mark is not sequenced and is an example of a more unconventional phrase that contains some tension.

A major influence on the results within approach one is the strength of the attraction to the pheromones. *Example 2* has identical settings as *example 1* except the attraction to the pheromones is decreased. Since the swarm is less compact and less bound to the pheromone there is a wider range of notes that can be generated and that are contained within one melodic phrase. This leads to a higher frequency of ascending and descending melodic phrases which can work very well. However, the randomly sequenced phrases get a more chaotic nature. *Example 3* has this same low attraction to pheromones but with a more compact swarm. The resulting phrases are similar to the phrases in *example 1* but are more diverse.

Example 4 illustrates what happens if the attraction of the pheromones is strong but the separation of the swarm is also strong. These two forces contradict each other which makes the swarm very active. Since the individuals are moving very fast they generate flurries of short notes which possesses a distinct character. *Example 4* also illustrates the effectiveness of the system over a minor chord progression.

3.3 Approach 2

Our second approach consists of three variations which all produce different musical results. The first variation is a strong chord-based pattern where the chord tones have a higher chance of being played than non-chord tones. It produces less pronounced musical phrases than the first approach since it is dependent of the target areas actually being activated. An example of the intended musical outcome can be found in *example 5* at the 21 second mark. More often sporadic quiet notes are played without a very defined melodic phrase. Better results are achieved when we make the swarm more volatile by decreasing the cohesion and increasing the separation. *Example 6* contains slightly more pronounced musical phrases.

The second variation has a higher density of target areas and uses a one-to-one mapping instead of a many-to-one mapping. This means a lot more target areas are activated and more defined phrases are generated. The melodic variation within these phrases is greater than the rhythmic variation. The melodic value of a target area changes for each chord but the rhythmic value remains constant. This is clearly audible in *example 7*. Again, the results are different if the swarm behaves more erratic, with low cohesion and higher separation versus a more condensed swarm. *Example 8* illustrates the effect of a more condensed swarm with more individuals. There is a higher frequency of quiet notes and identical notes appear more often. Although the results with the second variation are more promising than the first variation they still lack in variability, mainly due to the lack of diversity in rhythm.

Our third variation produces the most defined melodic phrases of the target area approaches. Its results come closest to the results achieved with the direct interpretation approach. Instead of the chord-based or probability-based variations the third variation is scale-based in combination with a high frequency of target areas being activated. *Example 9* illustrates the presence of ascending and descending melodic phrases in combination with more unpredictable phrases such as the one at the 1.03 mark. In contrast to the other variations, optimal results are achieved with a more condensed and stable swarm due to the linearity of the target areas. If we examine *example 10* we hear a greater frequency of erratic melodic phrases.

3.4 Approach 3

The third approach is a simplified version of the first approach with the addition of anticipation of the upcoming chord. In examining the examples we will focus on highlighting successful appearances of notes that lead to the next chord. A multitude of pronounced examples can be found in *example 11*. For instance, at the 8 second mark the A note is played over a G chord just before the harmony switches to the actual A minor chord. This extra quality present in approach 3 enhances the relationship between melody and harmony and creates a more dynamic interplay.

To illustrate why approach 3 is not interesting from a swarming perspective we included *example 12*. This example was recorded using a single individual. There is only a slight difference between this example and the previous example in the overall qualities of the output. This would not have been the case if swarming behavior actually made an impact on the system.

4 Discussion

4.1 Approaches and restrictions

Our three approaches all offer different mappings and restrictions on swarm behavior. We will now assess the effects of the choices we made for our basic implementation and for each approach individually. We also try to offer possible solutions to some shortcomings of our system.

In our first and third approach we chose to map the pitch to the y-axis because this is visually the most intuitive relationship. However, this fails to provide any meaningful correlation with swarm behavior. If we would have mapped pitch to the x-axis for instance the results would have been the same. Conceptually the mapping would have been stronger if pitch was related to an intrinsic aspect of the swarm, such as for instance speed or density. However, our use of attracting pheromones requires some effective way to modify the swarm space. By equating the position of pheromones and agents to the same pitch value we create a strong bond between input and output. This bond would have become less direct if notes from the harmony would somehow have to be translated to intrinsic swarm aspects. By breaking the conceptual mapping for one parameter we created an effective bridge between input and output.

The use of a sequencer in our first approach leads to more structured melodies but is unrelated to swarm behavior. Relating the sequence of the notes to some aspect of swarm behavior would be more meaningful. For instance, we could add an extra dimension to the swarm. Then, if the swarm was to move in a 3-dimensional space the sequence of notes could be determined by their position on the z-axis. However, in this case it would still be difficult to create a conceptual relationship between swarm behavior and sequence. We have also not been able to find such a conceptual relationship in relevant literature.

Our second approach does not reflect swarm behavior directly. Instead, it aims to create an environment in which a swarm can produce musical results. We are convinced this leads to valid and interesting results. However, one must be careful to keep the influence of the swarm relevant. It would for instance be an option to combine target areas with pheromones. This way both input and output influence each other, leading to conceptually stronger results. In the current state we think our approach is valuable because it contrasts so strongly with the first approach. We have no influence over the behavior of the swarm and the interpretation is very simple.

Even without a strong relationship to swarm behavior the results can be valuable for comparing approaches. For example, it could serve as an interesting control group in determining the concrete value of the swarm itself. The use of target areas does offer a wide range of variables we can control. For instance, experiments could be done by controlling

the lifespan, growth, count and movement of target areas. By exploiting these variables perhaps a conceptually stronger link could be made between the swarm and the target areas, perhaps inspired by birds hunting for prey or other biological phenomena.

Our third approach functions by adding major restrictions to the output and singling out one aspect, namely knowledge of the chord progression. We think it is a good addition to our previous approaches because this characteristic is not presence in our other approaches. As a result, it is important to focus on this aspect when comparing our approaches. Especially because the mapping strategy used is simplified version of our first approach.

4.2 Importance of the swarm

The musical output of our system is the result of data generated by the swarm and a translation of this data to music using a theoretical framework. The stronger the conceptual relationship between the behavior of a swarm and the mapping, the bigger the importance of the swarm itself becomes. In our introduction we posed the challenge of determining the specific musical qualities of swarm algorithms. We unfortunately have to exclude a definitive answer to this question from the scope of our research. To properly study the specific musical qualities of a swarm would require a separate study which would compare swarm algorithms with other approaches ranging from random generated data to other biologically inspired emergent algorithms.

We can however discuss the importance of the swarm in relation to the theoretical framework in our approaches. Our first approach makes the most use of swarm behavior. It uses multiple variables in translating individuals to notes such as individual speed, average speed and relative position. These variables are the product of the complex interplay between individuals dictated by their cohesion, separation and alignment which can only be the result of a dynamic system. For this reason, we conclude that swarm behavior is indeed a very important aspect within our first approach. Other types of data generation would at least have to approach swarm behavior in order to produce similar results. The only caveat is that if we increase the attraction of the pheromones a lot of these subtleties are lost. In the extreme case, with maximum attraction, individuals vibrate around a chosen pheromone and the swarm will produce only two notes. In such cases the musical output would be identical using randomly generated values.

The target area approach exploits swarm behavior in a less intrinsic way. Instead it relies more on chance. The use of target areas comes down to optimizing the placement of the target areas and the movement of the swarm in such a way that the chance of a successful melody being generated is the highest. As can be heard in the examples, a change in parameters of the swarm does not lead to very pronounced changes in the musical output. From this we can conclude that the placement of the target areas is the most important factor, not the behavior of the swarm. We hypothesize that with the target area approach, similar results could be achieved if we replace the swarm with a different form of data generation.

The role of the swarm in our third approach differs slightly from the first approach. Firstly, the swarm is not affected by pheromones so it exhibits more free behavior. Another important difference is that instead of taking a snapshot, the swarm is interpreted in real-time which mirrors swarm dynamics more directly. This means the swarm is very directly related to the musical output. However, the range of notes the swarm is able to produce is limited to chord tones only which can make the output sound similar to a random arpeggiator. Due to this restricted range of choices and because the swarm behavior does not interact with the harmony we think the role of the swarm is less critical than in our first approach.

4.3 Further work

Our system could be expanded by allowing it to support more notes than those of one specific key. We have chosen to focus on music where notes forming a melody and harmony only belong to one key. Typically, a human musician will start with a specific key when composing a melody and harmony. To keep a melody interesting, musicians can add notes from outside a specific scale to their melodies, like the use of chromatic passing tones to add color and tension. Specifically, in blues and rock music the flatted fifth is used very often between a fourth and a fifth to add expressiveness. In addition, it would be interesting to experiment with scales that have less emphasis on chord tones such as the pentatonic scale.

This also applies to harmony. A composer will often add color to their chords by adding extra intervals such as a seventh or a ninth. By confining our harmony to consist purely of three-tone chords we lose these subtleties. In addition, non-diatonic chords can introduce

more variability to chord progressions. For instance, playing the fourth chord as a minor chord instead of a major chord adds a tension which is resolved by returning to the first. Similarly, playing the sixth chord as a major chord resolves to a five chord. The melody can adapt to this by playing chord tones from these chords that will lie outside the major scale. Our system could gain expressiveness by implementing these subtleties.

Since the aim of our research is creating structured melodies using swarm algorithms it might prove insightful to perform a musical Turing test as suggested by Tim Blackwell. The structural qualities of generated melodies typically seen in melodies composed by humans would become quickly apparent if they were juxtaposed with human generated melodies. When performing such a test the melodies must be comparable. The same scales and chord sequences would have to be used as well as the same limited number of notes.

4.4 Applications

It is possible to imagine a system which would combine the positive aspects of all three approaches. A system that would use a strong conceptual mapping in a modified swarm space with knowledge of the chord progression. In the future, we may or may not develop this system, but this is what we see as possible applications of our research. We hope that artists and scientists interested in creating music with swarm algorithms can take some aspects from our systems and apply these to their own. In a broader sense, ideas and techniques discussed here could be applied to melody generation using other techniques than swarm algorithms.

We think that melody generating swarm algorithms could find a suitable place within broader music generation systems. For instance, generative music systems such as Brian Eno's generative music apps [10] could be enhanced with a swarm algorithm module. Little strands of melody could be interweaved with harmonies and textures created by other systems. Our system is especially suited for this as it is able to adapt to the harmony instantly. It could also offer a visual enhancement to such systems by offering an intuitive link between image and sound not present in machine learning or genetic algorithms for instance.

One especially interesting aspect of our target area approach is that it could be most easily be translated to real life swarms. For instance, a school of fish could swim in an aquarium filled with distance sensors. If a fish would enter a specific zone a musical note could be triggered similarly to one of our approaches.

Finally, in the library of examples a bonus example is included which highlights some of the results of the swarm if it is taken outside its strictly artificial context. The example consists of the swarm improvising a melody using approach 1. Instead of the synthesizer harmony, the harmony is played by a human musician on a guitar. The piano sound of the swarm is replaced with a trumpet sound. The intent of this example is to give a slight taste of a more natural application of the developed system.

5 Conclusion

We created three swarm algorithm systems that are able to generate structured melodies to some degree and with some consistency. The first approach constitutes of creating a detailed translation of swarm parameters to musical notes. It makes the most use of swarm behavior and can therefore be seen as the most promising outcome. The second approach consists of embedding target areas in the swarm space which trigger musical notes. The third approach is a variation of the first approach which adds restrictions and experiments with anticipating the next chord of the progression. These last two approaches offer interesting insights in creative ways of melody generation with swarm algorithms but function less effectively as standalone systems. This is due to a weaker conceptual relationship between the behavior of the swarm algorithm and the musical output. The musical output of the first approach is a reflection of the qualities of the swarm, while the other two approaches utilize the swarm more as a method of number generation. Swarm behavior is suited for the generation of melodies but further research needs to be performed to find out what specific factors contribute to this success. For instance, a comparative study between swarm-generated melodies and melodies generated by other algorithms could provide more insight in the specific qualities of swarm-generated melodies.

6 References

1. Papadopoulos A., et al. Assisted Lead Sheet Composition using FlowComposer. In: Proceedings of the 22nd International Conference on Principles and Practice of Constraint Programming. Toulouse, France. Sept 2016.

2. Reynolds C. Boids: Flocks. Herds and Schools--a Distributed Behavorial Model. 1986;

3. Schacher JC, Bisig D, Kocher P. The Map and the Flock: Emergence in Mapping with Swarm Algorithms. Computer Music Journal, Vol. 38 #3, 2014. p. 49-63.

4. Blackwell T. Swarm music: improvised music with multi-swarms. Artificial Intelligence and the Simulation of Behaviour, University of Wales. 2003.

5. Blackwell T. Swarming and Music. Evolutionary Computer Music. Springer London; 2007. p. 194-217.

6. Jones D. AtomSwarm: A Framework for Swarm Improvisation. Applications of Evolutionary Computing. Springer, Berlin, Heidelberg; 2008. p. 423–32.

7. Unemi T, Bisig D. Playing Music by Conducting BOID Agents - a Style of Interaction in the Life with A-Life. Proceedings of A-Life IX, Boston, MA, p. 546-50. Cambridge, MA: MIT Press.

8. Tavares T, Godoy A. Sonification of population behavior in Particle Swarm Optimization: Extended Material. Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. Amsterdam, p. 51-52. ACM.

9. Povel, D. Melody Generator: A Device for Algorithmic Music Construction. J. Software Engineering & Applications, 2010, 3, p. 683-95.

10. Eno, B. Generative Music Apps. http://www.generativemusic.com.

Appendix 1

Approach 1: direct interpretation

Example #	*Chords*	*Cohesion*	*Separation*	*Alignment*	*Attraction*	*# Individuals*
1	C - G - Am - F	0.5	0.5	0.2	0.04	7 - 14
2	C - G - Am - F	0.5	0.5	0.2	0.36	6 - 9
3	C - G - Am - F	0.9	0.1	0.8	0.3	7 - 12
4	Am - Em - F - Dm	0.1	0.9	0.5	0.04	12 - 15

Approach 2: target areas

Example #	*Chords*	*Cohesion*	*Separation*	*Alignment*	*Variation*	*# Individuals*
5	C - G - Am - F	0.8	0.1	0.5	first	4
6	C - G - Am - F	0.3	0.5	0.2	first	7
7	C - G - Am - F	0.5	0.5	0.2	second	4
8	C - G - Am - F	0.8	0.1	0.8	second	11
9	C - G - Am - F	0.5	0.5	0.2	third	4
10	C - G - Am - F	0.1	0.8	0.5	third	4

Approach 3: prescience

Example #	*Chords*	*Cohesion*	*Separation*	*Alignment*	*# Individuals*
11	C - G - Am - F	0.8	0.3	0.8	10
12	C - G - Am - F	0.5	0.5	0.5	1

Example #: the number of the example in the zip-file.

Chords: the chord progression used to generate the examples

Cohesion: the force with which the individuals are attracted to each other. Lower value means lower cohesion.

Separation: the force with which the individuals are repelled from each other. Lower value means lower separation.

Alignment: the force which aligns the general heading of the individuals. Lower value means less alignment.

Individuals: the number of individuals used to generate the example. This number can vary within an example.

Attraction: the force with which the individuals are attracted towards the pheromones in the direct interpretation approach. Lower value means stronger attraction to pheromones.

Variation: indicates the specific variation used within the target area approach as defined within the methods section.