A PDDL Representation for Contradance Composition

Richard G. Freedman and **Shlomo Zilberstein**
College of Information and Computer Sciences
University of Massachusetts Amherst
{freedman, shlomo}@cs.umass.edu

Abstract

Classical planning representations such as PDDL are primarily designed for goal-oriented problem solving, but some tasks such as creative composition lack a well defined goal. Structured performing arts, despite lacking a specific goal for their composition tasks, can be sufficiently expressed as goal-oriented problems for their performance tasks. Using contradance as an example performing art, we show how to represent individual contradances as plans such that the composition task can be compiled into a performance problem that can be expressed in PDDL. That is, by accounting for additional properties that are useful in their composition, the solutions to the performance task also serve as solutions to the composition task. We conclude with some example contradances derived using a classical planner under various composition conditions.

1 Introduction

As one of the earliest challenges in artificial intelligence, classical planning has often focused on automated problem solving where tasks have well defined goals. However, creative tasks such as artistic composition, ranging from writing music to dance and martial arts choreography, lack well defined goals outside of completing the work (to avoid an infinite loop of creation). Early research in creative artificial intelligence (Schmidhuber 2010) identified features such as constructing new things from simple patterns that cannot be easily expressed by past observations, but these features serve more as heuristics than actual approaches. Increased interest in creative machines has led to the formulation of the Lovelace Tests (Bringsjord, Bello, and Ferrucci 2001; Riedl 2015) and work in dynamic storytelling agents (Riedl and Young 2010; Amos-Binks 2017). We differentiate these tasks from the traditional ones as follows:

Definition 1 *A* goal-oriented task *in domain $\mathcal{D}$ is one for which a solution is any plan or policy π that yields a state satisfying the set of goal conditions. This includes finding optimal solutions as part of the task because any optimal solution is acceptable if more than one exists.*

Definition 2 *A* composition task *in domain $\mathcal{D}$ is one for which a solution is a plan or policy π that not only yields a state satisfying the set of goal conditions, but also satisfies specific intrinsic properties. These properties may include (but are not limited to) having particular action subsequences, following various rules describing π's structure, and expressing a desired message.*

In this work, we will investigate the creative task of contradance composition. Originating from Irish folk dancing, contradancing is a popular casual group dance in present times. The dancers form two long lines that usually break down into groups of four dancers. A caller announces a sequence of moves (called figures) that the dancers perform to reposition themselves within their group of four, sometimes mixing groups in more complicated sequences. The final figure in the sequence switches the positions of the two pairs of dancers to form new groups of four; this progression through the line continues as the sequence of figures loops until the song ends.

The steady progression and repositioning of dancers within their groups is well-structured with various mathematical properties (Peterson 2003; Copes 2003). Although these can be used to find the set of all possible contradances, composers select subsets of these contradances due to their artistic preferences and what they believe the dancers will enjoy. Thus *simply knowing how figures will alter the positions of dancers is not sufficient*. A random contradance generator created by a computer science professor (Frederking Publication Date Unavailable), which uses depth-first search to choose the next node to expand by a user-provided seed, even warns users that the dance is not guaranteed to feel right.

These systematic approaches all simply consider the rearrangement of dancers and select moves to accomplish these transitions. However, contradance composers have revealed that more than position is used when they create their sequences of figures (Dart 1995). Enumerating the state space with additional features typically leads to exponential scaling, but using a first-order logic representation of the state can help reduce these impacts for knowledge engineering. Furthermore, we can take advantage of other features of PDDL (McDermott et al. 1998; Fox and Long 2003; Gerevini et al. 2009) to properly formulate the progression as a classical planning problem such that off-the-shelf planners can find dances as sequences of figures. We begin with a background of the contradancing domain in Section 2. Section 3 uses these details to illustrate how to represent the states and actions. With the PDDL representation derived,

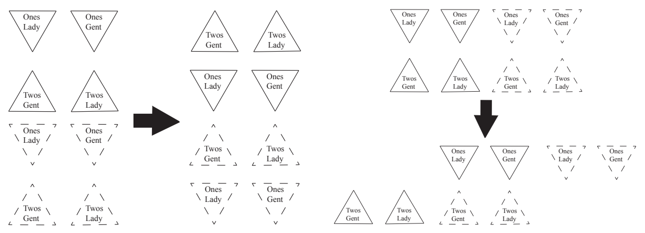


Figure 1: An illustration of progression for duple improper (left) and Becket formation (right). The couple that reaches the end changes between ones and twos to begin moving in the opposite direction with each progression.

we empirically present some of the contradances composed by off-the-shelf planners in Section 4 and discuss how this can be utilized and extended in Section 5.

2 Description of the Contradance Domain

A contradance is a well-structured community dance that has a formulaic procedure for set up and performance. Dancers are initially paired into *couples*, one *partner* in a couple is assigned the *gent* role and the other partner is assigned the *lady* role. The present-day interpretation of the roles is not indicative of who is dancing, but the lady is often located to the right of the gent and a few figures have different steps for each role working together as a couple. Depending on the choreography of the dance, couples are grouped together; the majority of dances have *duples* with two couples, and we will assume duples for our state space definitions in Section 3. In a duple, each couple faces each other where the couple facing the front of the dance hall (where the caller is located) is called the *ones* and the couple facing the back of the dance hall is called the *twos*. The ones gent and twos lady are called *neighbors* as are the ones lady and twos gent. All the duples are then joined to form two lines of dancers; it is called a *duple improper* when the neighbors are in the same line and a *Becket formation* when the partners are in the same line.

After setting up the dancers, the caller will announce a sequence of figures for everyone to perform. The steps in most figures only involve dancers within a duple so that each duple is dancing independently in parallel — some more complicated dances will have a *shadow* role for dancers between different duples who will interact with each other. However, the community still dances together even without a shadow due to the *progression*. The progression, resulting from the sequence of performed figures, moves the ones couple into the next duple closer to the front of the dance hall and the twos couple into the next duple closer to the back of the dance hall. See Figure 1 for an illustration of the progression and lines setup. After the progression, the sequence of figures is repeated using the newly formed duples; thus partners remain constant throughout the entire dance while neighbors change with each progression.

The length of time between progressions is sixty-four beats of music divided into four sets of sixteen beats. The music is divided into two sixteen-beat phrases that are each played twice during a single performance of the sequence of figures. Although this musical feature does not affect the dancers, the choreography uses this such that no figure is performed between two sets. That is, a figure must end when a musical phrase ends; this produces four partitioned subsequences of figures that each last sixteen beats. Figures vary in duration of beats from one beat to eight beats, which yields some of the variety between dances. To avoid monotony, most figures can only be repeated a specific number of times in a row. However, one figure is typically expected in every contradance: the swing. A swing is most often performed between partners or neighbors where the gent and lady hold waists/shoulders and spin clockwise about their center axis for eight beats, ending with the lady positioned to the right of the gent. Many contradances have one swing with the partner and another with the neighbor, but each swing is never done more than once per sequence of figures. After many progressions, the music eventually stops looping and ends the dance's performance.

3 Contradancing Representation in PDDL

The PDDL representation of any domain is $\mathcal{D} = \langle F, A \rangle$ where F is the set of fluents such that 2^F is the state space and A is the set of actions that can alter the states via add and delete effects $add\,(a \in A)\,, del\,(a \in A) \in F$ if their preconditions $pre\,(a \in A) \in F$ are satisfied. A problem in a given domain is represented as $\mathcal{P} = \langle \mathcal{D}, I, G \rangle$ where $I \in 2^F$ is the initial state and $G \subseteq F$ represents the goal conditions that must be satisfied for the task to be completed. When using first-order logic, we introduce the set of object types T into $\mathcal{D}$ and actual objects of type $t \in T$, $\mathcal{O}_t$, into $\mathcal{P}$ such that the fluents and actions are lifted over a tuple of parameters $params\,(x \in P \cup O) \in T^*$:

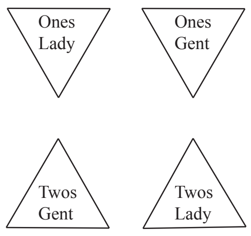
$$F = \bigcup_{p \in P} \bigotimes_{t \in params(p)} \mathcal{O}_t\ ,\ A = \bigcup_{o \in O} \bigotimes_{t \in params(o)} \mathcal{O}_t$$

where P is the set of propositions, O is the set of operators, and $*$ is the Kleene closure for any sequence of zero or more elements from a set.

For the contradance domain, we will follow the works in mathematics that view the layout of dancers in the dance hall as the state space and the figures as operators that alter this layout. Because a single caller dictates the figures and all the dancers follow these instructions, we view contradancing as a *centralized multi-agent problem*, which can be represented as a single agent problem where each action dictates what all the agents (dancers) do at once. Hence our type set T contains $dancer$, $location$, and $direction$ for each agent and the layout. We will also need T to contain $beat$ and set for temporal purposes when defining the operators.

Contradancing States

The dance hall will be laid out and connected in a manner similar to the traditional GRIDWORLD domain because contradancers are always lined up with each other before/after the performance of each figure. This gives us the propositions $adjacent$ and $same\_line$ where $params\,(adjacent) = (location, location, direction)$ and



```
(at_loc ones_lady 11)   (adjacent 11 12 left)   (same_line 11 21)
(at_loc ones_gent 12)   (adjacent 12 11 right)  (same_line 21 11)
(at_loc twos_gent 21)   (adjacent 21 22 left)   (same_line 12 22)
(at_loc twos_lady 22)   (adjacent 22 21 right)  (same_line 22 12)
(facing ones_lady 21)   (adjacent 11 21 back)
(facing ones_gent 22)   (adjacent 21 11 front)
(facing twos_gent 11)   (adjacent 12 22 back)
(facing twos_lady 12)   (adjacent 22 12 front)
```

Figure 2: A duple of dancers presented visually with its translation into PDDL.

$params\left(same\_line\right) = \left(location, location\right)$. For GRID-WORLD, directions are simply front, back, right, and left. Due to the lack of transitivity and symmetry between relations in PDDL, a problem's initial state must include all $2n\left(n-1\right)$ $same\_line$ fluents that are true to form the two lines of length n locations. Similarly, the problem must identify all pairs of locations for $adjacent$ with the directions being opposite. We present a visualization of a grid with a duple and these relations in Figure 2.

The dancers presented in the duple are positioned at locations using proposition at_loc where $params\left(at\_loc\right) = \left(dancer, location\right)$. Although this is sufficient for the mathematical representations, there are additional dancer features that a contradance composer keeps in mind to avoid awkward figures that the mathematical and automated approaches currently encounter. The simpler set of features identify roles because some figures require specific positioning of the gents and ladies; thus $params\left(role\_gent\right) = params\left(role\_lady\right) = params\left(role\_1s\right) = params\left(role\_2s\right) = \left(dancer\right)$ and $params\left(partner\right) = params\left(neighbor\right) = params\left(shadow\right) = \left(dancer, dancer\right)$ where the symmetry between these relations must again be defined.

The more complex feature to consider is based on the dancers' flow from the previous figure. If a dancer is moving forward and then told to interact with the dancer to the left by passing right shoulders, there will be some difficulty because the dancer needs to turn to the left and slow down enough to coordinate the shoulder passing — it would be more natural to pass by the left shoulder in this case due to the turn. Although computing a specific velocity in each direction can be challenging, we have found it sufficient to represent the flow by the direction that a dancer is facing; $params\left(facing\right) = \left(dancer, location\right)$ denotes the location in front of the specified dancer. This means a dancer may be facing at an angle if the location is not adjacent in the grid, which happens in several common figures.

Contradancing Initial and Goal States

In addition to the initial state providing the layout of locations in the dance hall, it must also set up the time (first beat in the first set), dancers, and their roles. In particular, because the lines may be formed by an arbitrary number of duples of dancers, it is only necessary to plan for the fewest number of duples who will perform the figures together, ignoring those dancing in parallel independently:

Proposition 1 *It is sufficient to define a contradance planning problem $\mathcal{P}$ containing $4\left(1+2k\right)$ agents/dancers where k is the number of duples between a given dancer and its respective shadow in the line, but it is only necessary to define $\mathcal{P}$ containing $4\left(1+s\right)$ agents/dancers where s is the specific number of shadows assigned to a dancer – usually $s \in \{0, 1\}$.*

Proposition 1 needs to consider k duples on either side of the couple because shadows become adjacent to their dancers through temporary progressions that take place within the four sets of sixteen beats. Thus the ones couple's shadows are located k duples in one direction while the twos couple's shadows are located k duples in the opposite direction because the ones and twos couples move in opposite directions during progression. Since all duples perform the same figures in parallel, it is also possible to control all the dancers at once regardless of how many duples are present in the problem. However, to reduce the overhead of keeping track of so many dancers, it is only necessary to represent one duple (for the actual choreography) and the specific dancers who serve as the shadows for each dancer in this duple (to check preconditions for figures involving shadows). Each shadow has a role in its own duple, and figures may manipulate the shadows as they affect the respective dancers of each role in the represented duple.

Proposition 2 *It is necessary and sufficient to define a contradance planning problem $\mathcal{P}$ containing $8\left(1+d\right)$ locations where $d = \left(1+2k\right)$ is the number of duples in Proposition 1.*

In Proposition 2, $4d$ of the locations are used for the two lines of d consecutive duples of dancers; this much of the dance hall must be present for the shadows to meet from their initial locations. The remaining $4\left(2+d\right)$ locations form a border around the two lines in order to accommodate *facing outwards*. The borders that are parallel to the lines do not have the $same\_line$ proposition hold true, but satisfy the respective $adjacent$ propositions. See Figure 3 for this initial state layout.

Although most the bordering locations are not guaranteed to be used in a given sequence of figures, at least a few of them are required for the goal condition of completing a progression. Figure 4 has PDDL translations of two initial and goal states for single duples without shadows (based on Figures 2 and 3). When starting from a duple improper, the couples who reach the ends of the lines should be facing

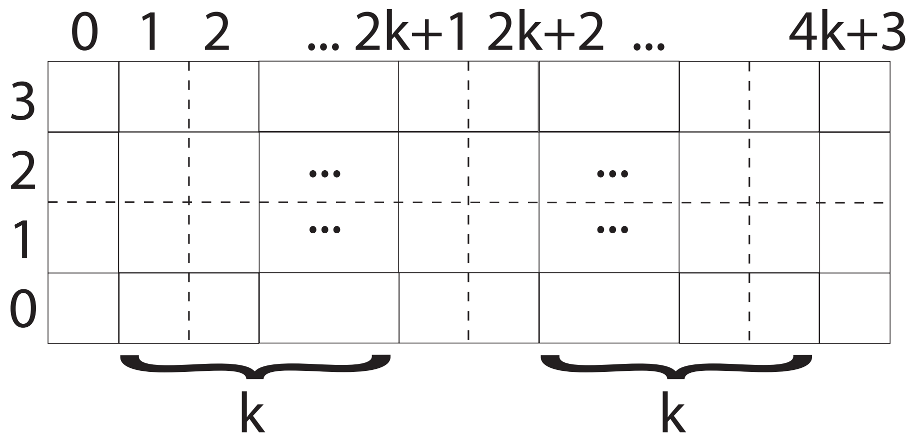


Figure 3: The grid layout of the dance hall described as the initial state. It is rotated 90 degrees counterclockwise so that the left direction is on the bottom. Thus rows 1 and 2 serve as the two lines while rows 0 and 1 and columns 0 and $4k$+3 are the border.

these border locations. When starting from a Becket formation and progressing clockwise, the couples who reach the ends of the lines should have the gent (who is to the left of the lady) at a location along this border. In addition to the changes in dancer location (although $facing$ changes for the dancers, their directions are preserved), the goal conditions should update the temporal information to the last beat of the last set (or the first beat of the set after the last, depending on implementation). This creates the task of *producing a progression in exactly the specified amount of time*, which is the goal of contradance performance.

Contradancing Actions

Given the layout of dancers throughout the dance floor, the figures will alter their locations and facing directions. Most operators represent a single figure whose parameters are the temporal beat assignment, dancers who will perform the figure, and their locations; this is usually the entire tuple, but there are figures where only two of the dancers move and the others remain still. However, dancers cannot perform a figure if they are not in the correct layout nor does a figure flow comfortably if they are facing the wrong directions with each other. This is where the preconditions are necessary to identify when a figure may be performed. The common features of operator preconditions for the contradance domain are:

- Enough beats remain in the set to perform the figure
- All the dancers in the parameters are different
- Each dancer is at the location specified in the parameters
- The locations are laid out correctly for the performance
- The dancers are facing the correct locations for flow

Likewise, the common components of operator effects (both for add and delete) for the contradance domain are:

- Increment the current beat by the duration of the figure
- Permute the dancers' locations if they change
- Change the dancers' directions if they change
- Increase the path cost if operator costs are used to guide the planner (see Section 4)

We call the set of operators representing figures $O_{figures}$, and the remaining operators are used to realign the temporal and spatial inforation $O_{realign} = O - O_{figures}$. The realignment operators do not have any cost and allow semantically equivalent representations to be used in the state space such as incrementing the set, which resets the beats from sixteen to zero to allow the next set of figures to begin, and repositioning the dancers as continuation of their flows (see Figure 5 for some examples). Although these operators could be embedded within the effects of figure operators, it is easier from a compositional perspective to observe the distinct transition in sets and movement of dancers. This also facilitates the caller's job as there is an explanation of what the dancers should do to position themselves for each figure's performance.

Counting Consecutive Figures If the number of repetitions is also considered as a constraint to reduce monotony in possible plans, then we must make several simple modifications. First the type 'repetitions' must be included in T. This allows us to include propositions of the form $consecutive\_\odot$ for each operator $\odot \in O_{figures}$ where $params\left(consecutive\_\odot\right) = \left(repetitions\right)$. Then each operator contains preconditions that ensure the maximum number of repetitions is not yet achieved for that figure, and the effects reset the repetition counts for all other operators while incrementing the repetition count for the performed figure.

Representation of Counting Although the use of numerical fluents (not to be confused with the domain fluents) has been considered a feature of PDDL since version 2.1 (Fox and Long 2003), many off-the-shelf classical planners limit their use to computing plan cost for optimization. So despite the ease of representation as integers with functions increment, assign, and compare to handle the beat, set, and rep-

```
(:init
 ... ;Set up the dance floor grid
 (at_loc ones_lady 11)
 (at_loc ones_gent 12)
 (at_loc twos_gent 21)
 (at_loc twos_lady 22)
 (facing ones_lady 21)
 (facing ones_gent 22)
 (facing twos_gent 11)
 (facing twos_lady 12)
 (current_set s0)
 (current_beat b0)
)
(:goal
 (at_loc ones_lady 21)
 (at_loc ones_gent 22)
 (at_loc twos_gent 11)
 (at_loc twos_lady 12)
 (facing ones_lady 31)
 (facing ones_gent 32)
 (facing twos_gent 01)
 (facing twos_lady 02)
 (current_set s4)
 (current_beat b0)
)
```

```
(:init
 ... ;Set up the dance floor grid
 (at_loc ones_lady 21)
 (at_loc ones_gent 11)
 (at_loc twos_gent 22)
 (at_loc twos_lady 12)
 (facing ones_lady 22)
 (facing ones_gent 12)
 (facing twos_gent 21)
 (facing twos_lady 11)
 (current_set s0)
 (current_beat b0)
)
(:goal
 (at_loc ones_lady 11)
 (at_loc ones_gent 01)
 (at_loc twos_gent 32)
 (at_loc twos_lady 22)
 (facing ones_lady 12)
 (facing ones_gent 02)
 (facing twos_gent 31)
 (facing twos_lady 21)
 (current_set s4)
 (current_beat b0)
)
```

Figure 4: PDDL representations of initial state and goal conditions for dancers starting from duple improper (left) and Becket formation (right). The goal conditions create a progression after the four sets of sixteen beats elapse.

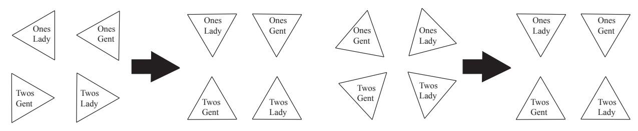


Figure 5: Realignment actions allow dancers to continue their flow for repositioning and match the preconditions of the next figure. The left image shows the flow of two dancers facing outwards to rotate towards the dancers on their left (who also rotate to face them). The right image shows the flow of dancers facing each other along the diagonals to continue moving until they face each other in the same line.

etition counts, it is not a feasible representation if we want to employ current planners to solve the contradance composition task. Therefore, we instead use conditional effects (part of the adl requirement) with the integers as domain constants of each type, and then enumerate all the precondition comparisons as a conjunction of negated equivalence checks as well as all the effect increments as a conjunction of conditional statements (hard-coding the increment). Figure 6 compares the numerical fluent and conditional effect PDDL representations.

4 Empirical Exploration of Composition

For initial evaluation of our formulation and representation, we encoded the contradancing PDDL domain with the six elementary figures detailed in Table 1. The swing is omitted because despite being a staple figure in any composition, its result is nondeterministic when considering flow — the lady ends to the right of the gent, but the direction is ambiguous depending on whether it is intended for a progression or part of the current set. We also included the realignment operator for incrementing the set after sixteen beats. When this domain was run in the state-of-the-art FastDownward Planner (Helmert 2006) with a problem that contained no shadows and started at the first beat of the first set in a duple improper formation, we received the following plan as the system's contradance composition:

```
Beat0, Set0: Do Si Do (Neighbors)
Beat4, Set0: Right and Left Through
Beat8, Set0: Right and Left Through
Beat12, Set0: Do Si Do (Neighbors)
Beat16, Set0: Update to Set1
---
Beat0, Set1: Right and Left Through
Beat4, Set1: Right and Left Through
Beat8, Set1: Do Si Do (Neighbors)
Beat12, Set1: Right and Left Through
Beat16, Set1: Update to Set2
---
Beat0, Set2: Right and Left Through
Beat4, Set2: Do Si Do (Neighbors)
Beat8, Set2: Right and Left Through
Beat12, Set2: Right and Left Through
Beat16, Set2: Update to Set3
---
```

```
(:action numeric_fluent_version
 :parameters (...)
 :precondition
 (and
   ;Must be within the time
   (< (current_beat) 13)
   ... ;Check locations, flow, and roles
 )
 :effect
 (and
   ;Add 4 beats to the set
   (increase (current_beat) 4)
   ... ;Update locations and flow
 )
)

(:action conditional_effect_version
 :parameters (... ?b - beat)
 :precondition
 (and
   ;Must be within the time
   (current_beat ?b)
   (not (= ?b b13))
   (not (= ?b b14))
   (not (= ?b b15))
   (not (= ?b b16))
   ... ;Check locations, flow, and roles
 )
 :effect
 (and
   ;Add 4 beats to the set
   (not (current_beat ?b))
   (when (= ?b b0) (current_beat b4))
   (when (= ?b b1) (current_beat b5))
   (when (= ?b b2) (current_beat b6))
   (when (= ?b b3) (current_beat b7))
   (when (= ?b b4) (current_beat b8))
   (when (= ?b b5) (current_beat b9))
   (when (= ?b b6) (current_beat b10))
   (when (= ?b b7) (current_beat b11))
   (when (= ?b b8) (current_beat b12))
   (when (= ?b b9) (current_beat b13))
   (when (= ?b b10) (current_beat b14))
   (when (= ?b b11) (current_beat b15))
   (when (= ?b b12) (current_beat b16))
   ... ;Update locations and flow
 )
)
```

Figure 6: The PDDL code for counting using numerical fluents (top) and conditional effects (bottom) when a figure has a duration of four beats.

Table 1: Contradance Figures Implemented

Name	**Duration (Beats)**	**Maximum Repetitions**
Circle to the Left	1	4
Circle to the Right	1	4
Do Si Do	4	1
Long Lines Forward and Back	8	1
Pass Through	2	3
Right and Left Through	4	2

```
Beat0, Set3: Do Si Do (Neighbors)
Beat4, Set3: Circle to the Left
Beat5, Set3: Long Lines Forward and Back
Beat13, Set3: Circle to the Left
Beat14, Set3: Pass Through (Progression)
Beat16, Set3: Update to Set4
```

The first three sets display very little variety and avoid circling to the left or right, which have the shortest duration of all the implemented figures. This decision can be explained by the fact that FastDownward is a goal-oriented optimal planner. That is, it will find the plan with least total action cost as the solution. For a formulation without defined/with uniform operator costs, this means that figures with longer duration will be preferred over figures with shorter duration because fewer figures are performed to reach the final beat of the final set. Thus the *composition cost* of a contradance operator $o \in O_{figure}$ is $cost\,(o)\,/duration\,(o)$.

Definition 3 *The* composition cost *of an action is the cost of selecting it for a plan rather than executing it. The composition cost should be lesser if it yields a greater reduction to the composition effort, has more desirable qualities for the plan requirements, etc..*

If it was possible to set up the dancers for the 'Long Lines Forward and Back' figure without performing 'Circle to the Left', then we would have seen it performed in the first three sets above because it has the cheapest composition cost $1/8$. However, the required 'Circle to the Left' would cost an additional action and offset the beat to odd parity where circling is the only figure that could return the beat to an even parity for completing the set. Thus it would cost three actions for ten beats, requiring at least two more actions to complete the set for a total of five actions. This costs more than four actions with four beats of duration each.

Because FastDownward can optimize over a total cost numeric fluent, we then re-evaluated our domain with assigned costs for each operator representing a figure. In particular, we identified composition costs for each figure using the formula above and then converted them to actual operator costs. The first case assigned all figures a uniform composition cost: $cost\,(o) = duration\,(o)$. Thus all solutions now have the same total path cost and the returned plan will depend on which solution is found first. The resulting plan found was:

```
Beat0, Set0: Right and Left Through
Beat4, Set0: Do Si Do (Neighbors)
Beat8, Set0: Right and Left Through
Beat 12, Set0: Do Si Do (Neighbors)
Beat 16, Set0: Update to Set1
---
Beat0, Set1: Right and Left Through
Beat4, Set1: Do Si Do (Neighbors)
Beat8, Set1: Right and Left Through
Beat 12, Set1: Do Si Do (Neighbors)
Beat 16, Set1: Update to Set2
---
Beat0, Set2: Right and Left Through
Beat4, Set2: Do Si Do (Neighbors)
Beat8, Set2: Right and Left Through
Beat 12, Set2: Do Si Do (Neighbors)
Beat 16, Set2: Update to Set3
---
Beat0, Set3: Right and Left Through
Beat4, Set3: Do Si Do (Neighbors)
Beat8, Set3: Right and Left Through
Beat12, Set3: Circle to the Left
Beat13, Set3: Circle to the Right
Beat14, Set3: Pass Through (Progression)
Beat16, Set3: Update to Set4
```

Lastly, to introduce a composer's preference for some figures over others, we assigned composition cost to be the duration $cost\,(o) = duration\,(o)^2$ so that figures with shorter duration have lesser cost (the opposite of the uniform operator cost case) and received the following composition (we only present the first and last set due to space — the first three sets are identical):

```
Beat0, Set0: Circle to the Left
Beat1, Set0: Circle to the Right
Beat2, Set0: Circle to the Left
Beat3, Set0: Circle to the Right
Beat4, Set0: Circle to the Left
Beat5, Set0: Circle to the Right
Beat6, Set0: Circle to the Left
Beat7, Set0: Circle to the Right
Beat8, Set0: Circle to the Left
Beat9, Set0: Circle to the Right
Beat10, Set0: Circle to the Left
Beat11, Set0: Circle to the Right
Beat12, Set0: Circle to the Left
Beat13, Set0: Circle to the Right
Beat14, Set0: Circle to the Left
Beat15, Set0: Circle to the Right
Beat16, Set0: Update to Set1
---
...
---
Beat0, Set3: Circle to the Left
Beat1, Set3: Circle to the Right
Beat2, Set3: Circle to the Left
Beat3, Set3: Circle to the Right
Beat4, Set3: Circle to the Left
Beat5, Set3: Circle to the Right
Beat6, Set3: Circle to the Left
Beat7, Set3: Circle to the Right
Beat8, Set3: Circle to the Left
Beat9, Set3: Circle to the Right
Beat10, Set3: Circle to the Left
Beat11, Set3: Circle to the Right
Beat12, Set3: Circle to the Left
Beat13, Set3: Circle to the Right
Beat14, Set3: Pass Through (Progression)
Beat16, Set3: Update to Set4
```

Using a diverse planner (Nguyen et al. 2012; Roberts, Howe, and Ray 2014) to solve these problems would be more ideal for composition tasks because there are a variety of plans with the same total cost that should be considered rather than just the first one found. Most planners have a deterministic procedure for tie-breaking during the generation of successor states; hence the plans found above are repetitive even with the consecutive figure constraints.

5 Discussion

Unlike goal-oriented tasks where the problem simply needs to be solved, composition tasks need to solve the problem with stylistic preferences. Using the performing art of contradance, we defined a state space and actions that force off-the-shelf classical planners to find sequences that are not only solutions to completing a performance, but exhibit desired intrinsic qualities such as respecting the flow of dancers, avoiding too many consecutive repetitions of figures, and including more preferred figures via composition cost. We believe that this compilation process may be used to solve other composition tasks via their goal-oriented counterparts. There are many potential applications for this work including the derivation of PDDL for similar composition tasks like square dancing, developing tools that allow human composers to receive recommendations for partially-composed dances with respect to their creative interests, and creating novel contradance figures to make unique dance patterns work (similar to Zook and Riedl's (2014) approach for developing game mechanics). These aspects will motivate and guide future research. We will also explore more complex composition cost formulas and experiment with other planners, such as diverse planners (Nguyen et al. 2012; Roberts, Howe, and Ray 2014), to investigate how they approach solving the compiled composition task.

Acknowledgements

We thank the anonymous reviewers for their feedback. This work was supported in part by NSF grant 1405550.

References

Amos-Binks, A. 2017. Problem formulation for accommodation support in plan-based interactive narratives. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence: Papers from the Doctoral Consortium.*

Bringsjord, S.; Bello, P.; and Ferrucci, D. 2001. Creativity, the Turing Test, and the (better) Lovelace Test. *Minds and Machines* 11(1):3–27.

Copes, L. 2003. Counting contra dances: From a talk given for the mathematical association of america. http://www.larrycopes.com/contra/MAA.html. [Online; posted after January-2003].

Dart, M. M. 1995. *Contra Dance Choreography: A Reflection of Social Change*. New York, New York, USA: Garland Publishing, Inc.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20(1):61–124.

Frederking, R. E. Publication Date Unavailable. Dr. Bob's random contra dances. http://www.cs.cmu.edu/∼ref/random-contras.html. [Online; last accessed 18-March-2017].

Gerevini, A. E.; Haslum, P.; Long, D.; Saetti, A.; and Dimopoulos, Y. 2009. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence* 173(56):619–668. Advances in Automated Plan Generation.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26(1):191–246.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL – The planning domain definition language – Version 1.2. Technical Report CVC TR-98-003, Yale Center for Computational Vision and Control, New Haven, CT, USA.

Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* 190:1–31.

Peterson, I. 2003. Contra dances, matrices, and groups. Science News, https://www.sciencenews.org/article/contra-dances-matrices-and-groups. [Online; posted after 5-March-2003].

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Riedl, M. O. 2015. The Lovelace 2.0 Test of artificial creativity and intelligence. In *Proceedings of the AAAI Workshop: Beyond the Turing Test*.

Roberts, M.; Howe, A.; and Ray, I. 2014. Evaluating diversity in classical planning. In *Proceedings of the Twenty-Fourth International Conference on Planning and Scheduling*, 253–261.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development* 2(3):230–247.

Zook, A., and Riedl, M. O. 2014. Automatic game design via mechanic generation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, 530–536. Québec City, Québec, Canada: AAAI Press.