

## RESEARCH ARTICLE

# Multi-Objective Deep Network-Based Estimation of Distribution Algorithm for Music Composition

JAE-HUN JEONG<sup>1</sup>, EUNBIN LEE<sup>2</sup>, JONG-HYUN LEE<sup>1,3</sup>,  
AND CHANG WOOK AHN<sup>1,2</sup>, (Member, IEEE)

<sup>1</sup>Research and Development Center, CreativeMind Inc., Suwon 16512, Republic of Korea

<sup>2</sup>School of Artificial Intelligence, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Republic of Korea

<sup>3</sup>Research Center for Convergence, Sungkyunkwan University, Suwon 16419, Republic of Korea

Corresponding authors: Jong-Hyun Lee (ljh08375@skku.edu) and Chang Wook Ahn (cwan@gist.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; The Ministry of Science and ICT) (No. NRF-2021R1A2C3013687, No. NRF-2020R1C1C1009720, No. NRF-2021R1F1A1064375).

**ABSTRACT** In the field of evolutionary algorithm music composition, most of the current researches focus on how to enhance environmental selection based on multi-objective evolutionary algorithms (MOEAs). However, the real music composition process defined as large-scale multi-optimization problems (LSMOP) involve the number of combinations, and the existing MOEA-based optimization process can be challenging to effectively explore the search space. To address this issue, we propose a new Multi-Objective Generative Deep network-based Estimation of Distribution Algorithm (MODEDA) based on dimensionality reduction in decision space. In order to alleviate the difficulties with dimensional transformation, we propose a novel solution search method that optimizes in the transformed space and ensures consistency between the pareto sets of the original problem. The proposed algorithm is tested on the knapsack problems and music composition experiments. The experimental results have demonstrated that the proposed algorithm has excellency in terms of its optimization performance and computational efficiency in LSMOP.

**INDEX TERMS** Evolutionary music composition, multi-objective optimization, generative deep networks, estimation of distribution algorithm, artificial intelligent music composition.

## I. INTRODUCTION

Evolutionary algorithms (EAs), inspired by the natural evolution, have achieved remarkable records in various fields of computational optimization and machine learning. Due to the domain-independent nature, they have been successfully applied to various industrial problems, even in the domains of art and music [1]. In particular, EA has established the field of evolutionary music composition, which diverts from data-based music research and exploits the search and optimization process and the method of designing fitness functions applying music theory. For instance, [2] proposed a method of creating a melody by designing a fitness function based on the intervals of the melody. In [3] and [4], the users' subjective feedback has reflected the fitness of generated melodies. Moreover, the generation of manmade-like

rhythms was tried by a genetic algorithm (GA) [5]. Existing methods have limitations in finding solutions that satisfy various perspectives, and better music should satisfy human subjective aesthetic perspectives [6]. To consider conflicting musical variables, a variety of multi-objective evolutionary algorithms (MOEAs) have been proposed. For instance, [6] conducted the melody harmonization using multi-objective genetic algorithms, and [7] and [8] generated a number of melodies by simultaneously optimizing multiple conflicting fitness functions.

Many musical compositions in the real world involve hundreds or even thousands of decision variables and a number of conflicting musical goals, due to the infinite combination of elements that make up music, such as rhythm, melody, and chord. These problems are defined as large-Scale multi-objective optimization Problems (LSMOP) [9]–[11], which are the main focus of recent evolutionary research. The existing multi-objective evolutionary algorithm (MOEA) method

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang .

does not effectively explore the search space for LSMOP problems with large search spaces and decision variables [12]. In general, MOEA tends to converge early to a local optimum, otherwise it may converge to a relatively large area [13]. In order to solve LSMOP, most existing MOEAs have been improved as a way to strengthen the strategy of choosing solutions for the next generation [14]–[16]. However, finding the optimal solution in a large search space by increasing search efficiency has limitations in solving the LSMOP.

In this paper, we propose a new Multi-Objective Generative Deep network-based Estimation of Distribution Algorithm (MODEDA) to effectively solve LSMOP problems in music composition. Our proposed algorithm is based on problem conversion through dimension reduction. Furthermore, we conducted an evolutionary research with multi-objective optimization to simultaneously generate a number of melodies that encompass a variety of listener's preferences. In terms of stability and tension, a multi-objective fitness formulation in the phase of evaluation is investigated to handle the multidimensional, conflicting nature of musical features aptly. In addition, a new deep network-based estimation of distribution algorithm is combined with a multi-objective approach to effectively traverse the search space of music composition, thereby generating diverse sweet melodies at once.

The rest of this paper consists of the following order: Section II offers some background knowledge on this work, including evolutionary multi-objective optimization and music terminology. Section III describes a new deep network-based multi-objective estimation of distribution algorithm, and introduces the proposed multi-objective approach to melody composition. Section VI verifies the proposed algorithm and demonstrates composition results with a summary. Finally, Section V describes the conclusion and future research.

## II. BACKGROUND

### A. MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization problems (MOPs) represent optimization problems with the objectives of conflicting relationships [17]. Typical MOPs are mathematically defined as follows [18]:

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T, \\ \mathbf{x} &= (x_1, x_2, \dots, x_n)^T, \\ \text{subject to } f_j(\mathbf{x}) &\leq f_j(\mathbf{x}^*), \quad \forall j = \{1, \dots, m\}, \\ f_j(\mathbf{x}) &= f_j(\mathbf{x}^*), \\ &\text{for at least one objective function} \end{aligned} \quad (1)$$

Multi-objective optimization finds the best set of solutions in the decision space, and this set of solutions is called the pareto optimal solution. Pareto-optimal front approximates these pareto-optimal surfaces in the target space. where  $(f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  are  $k$  objectives functions,  $x \in S$ ,  $(x_1, x_2, \dots, x_n)$  are the  $n$  optimization parameters,

and  $S \in R^n$  is the solution or parameter space.  $\mathbf{x}^*$  is said to be a pareto optimal solution of MOP [19], [20].

### 1) MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAS)

Evolutionary algorithms (EAs) find a solution by evolving a group of candidate solutions during repeated operations of fitness evaluation, selection, crossover and mutation. The EA approach can discover and return diverse non-dominated solutions over wider regions in search space simultaneously. Many real-world multi-optimization problems are challenged by a variety of characteristics and difficulties; non-linearity, the number of constraints and variables, complex relations among variables, several conflicting objectives, and so on [21], [22].

To efficiently address MOPs, various Multi-Objective Evolutionary Algorithms (MOEAs) based on EA have been proposed. Among them, the dominance-based-MOEA type solved the problem of MOP by approximating the pareto optimal set by distinguishing and selecting promising candidate solutions [23]. Typical, Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) [14] constructs an initial population through evaluation, and the solutions with the higher front order are selected in rank order until the original population size is reached. It performs non-dominate sorting and crowding distance comparison to select individuals and create new offspring through crossover and mutation. Pareto envelope-based selection algorithm II (PESAI) [24] and the improved strength Pareto EA (SPEA2) [25] also maintain non-dominant solutions and take a way to remove dominant solutions from the population.

### 2) LARGE-SCALE MULTI-OBJECTIVE OPTIMIZATION (LSMOP)

In traditional MOEA, regeneration tasks are typically based on stochastic mechanisms (e.g., crossover or mutation), so algorithms cannot be explicitly learned in the environment (e.g., fitness environment). Instead of creating sub-solutions through crossover or mutation of parent solutions, Estimation of Distribution Algorithms (EDA) builds and samples explicit probabilistic models of promising candidate solutions to explore the decision space of potential solutions [26]. Typical EDA procedures are given in Algorithm 1.

Typically, EDA based on Restricted Boltzmann Machines (RBM-EDA) creates a solution for the next generation by probabilistic sampling of late factors through a trained generation model [27]. EDA based on Deep Belief Network (DBN-EDA) is a model of Deep Belief Network with multiple layers of RBMs that create a new candidate solution by correcting weights through unsupervised learning [28]. EDA-based evolutionary algorithms have shown promising performance on a number of Multi-objective Optimization Problems (MOPs), but LSMOP problems dramatically degrade as the number of decision variables increases, and are computationally expensive to sampling multiple objectives [29].

Recently, many studies have been conducted based on problem conversion through generation models to solve this LSMOP [30]. [31] designed a GAN-based MOEA that produces offspring from GANs. The method samples promising candidate solutions that are useful for multiple optimizations using adversarial learning mechanisms from GANs, and efficiently learns high-dimensional distributions with limited training data. [32] uses a generation model to modify and maintain various solutions in a low-dimensional space based on Pareto’s optimal solutions that are uniformly distributed in manifolds in low-dimensional space. Instead of processing decision variables, [33] proposes a trend prediction model and a generating-filtering strategy, called LT-PPM, to improve search efficiency by maintaining population diversity.

**Algorithm 1** Pseudocode of EDA

**Procedure** Estimation of Distribution Algorithm  
**begin**

    Initialization( $D_0$ )

    Evaluate( $D_0$ )

**while** Stopping criterion is reached **do**

$D_1^s \leftarrow$  Select  $N$  individual from  $D_{l-1}$

$p_l(x) \leftarrow$  Estimate the probabilistic model from  $D_1^s$

$D_l \leftarrow$  Sampling  $M$  individuals from  $p_l(x)$

        Evaluate( $D_l$ )

**end while**

**end**

**B. VARIATIONAL AUTO-ENCODER**

In principle, the auto-encoder assumes that the latent space is deterministic, while the latent variable is stochastic in the variational auto-encoder (VAE) [34]. As a result, the auto-encoder is learned to reproduce precisely the same data of input. In other words, the latent space of the auto-encoder denotes the compressed representation of input data, which may not be continuous. From the viewpoint of the generative model, therefore, the auto-encoder does not work appropriately for generating new data through arbitrary latent variables.

However, the variational auto-encoder forces latent variables from the encoder to follow the Gaussian distribution so that the data is distributed continuously in the latent space. Due to the continuity of the Gaussian distribution of latent variables, generating new data that is not observed in the training data is very effective. The input data  $x$  is represented by latent variables in the latent space  $z$  through the encoder, then  $x$  is reconstructed through the decoder. The model is defined as follows:

$$p_\theta(z) = N(z; 0, I) \tag{2}$$

$$p_\theta(x|z) = N(x; \mu(z), \sigma(z)I) \tag{3}$$

$$q_\phi(z|x) = N(x; \mu(x), \sigma(x)I) \tag{4}$$

**TABLE 1.** Classification of non-chord tone.

| Non-Chord Tone | Approached by | Left by                    |
|----------------|---------------|----------------------------|
| Passing Tone   | Step          | Step in same direction     |
| Neighbor Tone  | Step          | Step in opposite direction |
| Appoggiatura   | Leap          | Step                       |
| Escape Tone    | Leap          | Leap in opposite direction |
| Anticipation   | Static note   | Same note                  |
| Retardation    | Static note   | Step up                    |
| Suspension     | Static note   | Step down                  |

The loss function for learning variational auto-encoder is expressed as

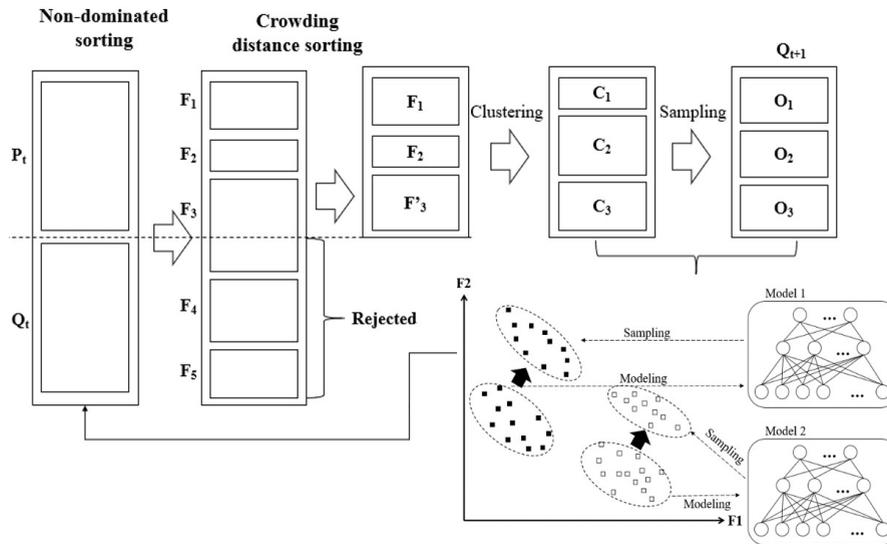
$$L = -E_{z \sim q(z|x)} [\log p(x|z) + KL(q_\phi(z|x) \parallel p_\theta(z))] \tag{5}$$

The first term is an expected negative log-likelihood of the training dataset, which means a reconstruction error. The expectation is taken with respect to the encoder’s distribution over the representations by taking a few samples. The second term is the Kullback-Leibler divergence between the distribution of encoder  $q_\phi(z|x) p_\theta(z)$ . This KL divergence calculates how much information is lost when using  $q$  to represent *a priori* over  $z$  and thus encourages its values to follow a Gaussian distribution. In general, a reparameterization trick is used to effectively perform inference operations using a backpropagation mechanism in the training phase of deep networks [35].

**C. MUSIC THEORY**

Most tonal music creates the atmosphere of the melody through the process of adding and relieving tension. There are various musical factors that create a sense of tension and stability through relaxation: harmony, dynamics, timbre, rhythm, melody, and even through the structure of a piece or song. Typically, there is a harmonic tension, which is a dissonant chord and chord progressions. Traditional harmony always requires a consonant chord, but harmonic tensions can be created by delaying and modulating the completion of this dominant chord. Most of the unstable chords consists of non-harmonic tones rather than chord tones (the 1st, 3rd, 5th, 7th note of the scale). Most of these embellishing tones of dissonances occur in weak beat (the 2nd and 4th beats duration 4/4 time), do not return directly to the chord tones, and naturally generate some tension.

Typically, dissonant tones are divided into three categories as follows: passing tones and neighbor tones including step-wise motion, appoggiatura and escape tones including leap motion. Each can be classified into suspension, retardation, and antique notes approached by static notes. Additional explanations for this are specified in detail in Table I.



**FIGURE 1.** Multi-objective generative deep network-based estimation of distribution (MODEDA) framework.

---

**Algorithm 2** MODEDA: Multi-objective generative Deep network-based Estimation of Distribution Algorithm

---

- 1: **Initialization** : At generation  $g := 0$ ; randomly generate  $2N$  initial population  $Pop(0)$
  - 2: **Evaluate** : Perform non-dominated ranking and crowding distance over the population
  - 3: **Selection** : Select  $N$  individuals based on binary tournament selection
  - 4: **Clustering** : Divide the selected population (according to objective values) into  $k$  clusters by the  $k$ -mean clustering
  - 5: **Modeling** : In each cluster, build an isolated network ( $k$  networks in total) based on VAE and perform a training process
  - 6: **Reproduction** : Generate a new set of  $N$  solutions,  $P$ , from the networks using a sampling method
  - 7: **Archiving** : Select  $N$  individuals from  $P \cup Pop(g)$  in order to obtain  $Pop(g + 1)$
  - 8: **Stopping Criteria** : If stopping criteria are met, Stop. Otherwise,  $g := g + 1$ , and go to Step 2
- 

### III. MULTI-OBJECTIVE DEEP NETWORK-BASED EDA FOR MUSIC COMPOSITION

Recent research on multi-objective optimization focuses on solving complex LSMOPs. In particular, since music composition consists of an infinite combination of elements such as rhythm, melody, and harmony, solving high-dimensional problems is also important in the field of music research.

In this sense, we propose a Multi-Objective generative Deep network-based Estimation of Distribution Algorithm (MODEDA) to solve multi-objective optimization problems of higher dimensions effectively. MODEDA is designed to replace the modeling and sampling procedure of multi-objective EDA [36] with the Variable auto-encoder (VAE) [34], in order to discover a Pareto optimal solution set based on dimensionality reduction. The VAE model compresses information about key variations from a higher-dimensional space to a lower-dimensional space, which aims to drive the population to an promising optimal solution.

We considered two problems when reducing the dimension of the decision space through VAE [29]. The first is to optimize in the transformed space and then solve the original problem. The second is to ensure consistency between

the Pareto set of transformed problems and the Pareto set of original problems. To solve two problems, MODEDA proposes a new search method. First, the new solution was optimized and explored in the decision space converted to low dimensions through the VAE model. It was then mapped to a solution set in the decision space, evaluated for suitability by the objective function, and finally created a solution set that guarantees consistency in the pareto front solution set. The detailed procedures of MODEDA are explained in the following subsections.

#### A. ALGORITHM FRAMEWORK

The proposed multi-objective generative deep network-based estimation of distribution algorithm (MODEDA) can adaptively deploy any operator of standard MOEAs [21], such as ranking, selection, or archiving, by replacing crossover and mutation with a specific generative deep network. MODEDA framework is illustrated in Figure 1. In this study, non-dominance ranking, crowding distance, and binary tournament selection that are similar to the operators in NSGA-II [14] are employed. Moreover, the  $k$ -means clustering is used to isolate the data group that conflicts with each other.

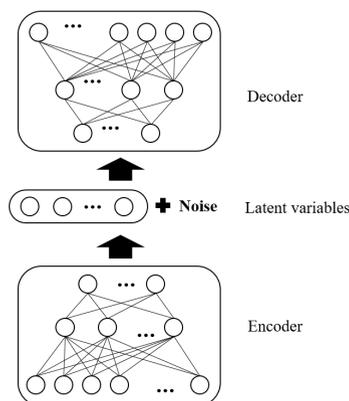


FIGURE 2. Sampling method for exploring new solution based on parent.

The overall procedures are described in **Algorithm 2**. Input layer’s units in the network are mapped to a set of variables in the cost function; the number of variables in the fitness function denotes the size of the input layer in VAE. The encoder part of the VAE compresses the search space to a low dimension to learn the critical dependencies of the training dataset, and then converts it into a latent space of the Gaussian distribution. The decoder part reconstruct the latent variable from the input data, and then to accommodate the new latent variable to generate new solutions. Each solution remapped the search space (N) and the newly optimized solution (N) in the Pareto optimal subspace to the original search space. The final solution set (2N) was fitted by the objective function in the search space.

The proposed algorithm utilizes multiple models along with the result of *k*-means clustering that decomposes a given problem [21], [37]. In general, the user determines the value of *k*; in this study, *k* is set to the number of objectives. The modeling and sampling process in Figure 1 is described as an example when the number of objectivities is 2. Thus, the whole population is divided into two groups, and a VAE model is trained in each cluster independently. Then, the models create  $\frac{n}{k}$  offspring by sampling the VAE model.

Our sampling method generates a new solution by adding a Gaussian noise to the parent population. To be more exact, VAE is able to represent data in a continuous probability distribution space, but the value itself is a black box. Thus, we add a Gaussian noise of  $N(0, 1)$  to 25 of the latent variables, which are obtained by entering the existing parent into the encoder layer. When generating new data from the trained VAE model, random latent variables are generally used as the input of the decoder. The Gaussian distribution of the trained VAE is unknown, and there are only a limited number of chances of sampling in the evolutionary process.

As shown in Figure 2, in order to effectively conduct the sampling process for discovering a non-dominated solution under these constraints, we use a modified latent variable of the training data. Random Gaussian noise  $N(0, 1)$  is added to some part of the latent variable, which is obtained from the encoder by putting the training data as the input. Then the

decoder generates new data from the modified latent variable. In this way, we can find more effectively new non-dominated solutions by generating data close to parents in the latent space.

In other words, MODEDA is able to find a new non-dominant solution that improves scalability in LSMOP more effectively by optimizing new solutions in low-dimensional search space and mapping them in their original space, creating solutions close to parents in the latent space. The complexity of the network would increase with the number of visible and hidden units. Since the modeling process is conducted at every generation, models need to be kept simple. Therefore, the number of hidden units is set to as small as possible, as long as the probability is representative. An extensive modeling process is not necessary since the exact distribution of the selected population does not represent the distribution of optimal solutions.

### B. MUSIC COMPOSITION PROCESS

The proposed MODEDA is applied to compose various, pleasant melodies in conflicting aspects of stability and tension. The music composition process is as follows. First, a chord progression and a rhythm are chosen by the expert knowledge or the existing materials. Then, our proposed MODEDA composes melodies with a dense and wide distribution in two conflicting musical aspects of stability and tension. Lastly, the user can choose a melody of desired preference without listening to all the outcomes.

#### 1) REPRESENTATION

The progress of each note constituting the melody is expressed as data having a sequence. First, each note was converted into a midi input to express pitch. Values were mapped according to the keyboard position based on the middle C key; 'C3' (i.e., 60) and 'D3' (i.e., 62). Then, the duration of the note can be represented by the length of each note. For example, a quarter note is mapped into '1.0', and a half note is mapped into '2.0'. Then, to express the rhythm, we mapped it to the duration value, which is the length of each note. Moreover, a rest note is encoded as '-1'. In this way, the initial population was built by converting pitch and rhythm information for one note into integer encoding and mapping it to chromosomes [8].

#### 2) MULTI-OBJECTIVE FORMULATION

In general, there are two conflicting factors in musical melodies; stability and tension. In principle, they are made up of several musical elements; stability is comprised of chord tone, step motion, and descending line, and tension consists of non-chord tone, skip motion, and ascending line.

From the two conflicting musical perspectives of stability and tension, the fitness function was designed as follows:

$$\arg \max Fitness(p, r) = \{F_{stability}(p, r), F_{tension}(p, r)\}. \quad (6)$$

For a given melody comprised of  $p$  (i.e., a pitch sequence) and  $r$  (i.e., a given rhythm sequence),  $F_{stability}(p, r)$  and  $F_{tension}(p, r)$  measure its stability and tension, respectively.

At first,  $F_{stability}(p, r)$  is defined as

$$F_{stability}(p, r) = \alpha CT(p, r) + \beta SM(p) + \gamma DL(p) - \delta P(p, r). \quad (7)$$

Here,  $CT(p, r)$  computes a fraction of chord tones; it returns a higher value in [0.0, 1.0] when the melody contains more chord tones.  $SM(p)$  measures a fraction of step motions, and  $DL(p)$  evaluates some of the three consecutive notes in descending order. The weight parameters of  $\alpha$ ,  $\beta$ , and  $\gamma$  are set as values of 3.0, 2.0, and 1.0, respectively, depending on the effect of melody stability. Moreover,  $P(p, r)$  is a penalty function that is described in Equation (9), and a large value (e.g., 100) has been assigned to the parameter  $\delta$  due to its importance in generating a harmonized melody.

Second,  $F_{tension}(p, r)$  is defined as

$$F_{tension}(p, r) = \alpha NT(p, r) + \beta LM(p) + \gamma AL(p) - \delta P(p, r). \quad (8)$$

In contrast to  $CT(p, r)$ ,  $NT(p, r)$  represents the fraction occupied by the non-chord tone of the melody and returns a higher value for the melody containing more non-chord tones.  $LM(p)$  evaluates a ratio of skip motions (i.e., leap motion), and  $AL(p)$  measures a fraction of three successive notes in ascending lines (as opposed to  $DL(p)$ ). The values of  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are equal to those used in Equation (7).

Meanwhile, there are some properties that possibly lead to an inappropriate melody; unresolved tension, restraint in strong beats, misused non-chord tones, and big leap motions. Such undesirable ones can be eliminated by employing a penalty term,  $P(p, r)$ .

The penalty term is defined as

$$P(p, r) = MNC(p) + AN(p) + ON(p, r) + BL(p). \quad (9)$$

Here,  $MNC(p)$  resolves some misused non-chord tones in a melody,  $AN(p)$  derives avoid notes over the strong beats not to be present in a melody,  $ON(p, r)$  makes the portion of measures (bars) that contain non-chord tones more than 50%, and  $BL(p)$  forestalls the inclusion of over-leap ( $\geq 9th$ ) intervals. More detailed information on all the terms of the fitness function including the penalty term can be found in [8].

#### IV. EXPERIMENTAL STUDY

We conducted two experiments to evaluate the performance of the proposed MODEDA. The first compares the scalability performance of our proposed model in the LSMOP problem compared to NSGA-II, RBM-EDA, and DBN-EDA through 10 knapsack experiments according to the expanding dimension. Second, through music generation experiments, the performance of the proposed MODEDA algorithm was compared with the NSGA-II algorithm to produce a well-converged solution to two conflicting objective functions with a broad, dense set of pareto solutions.

#### A. MULTI-OBJECTIVE KNAPSACK PROBLEM EXPERIMENT

The multi-objective knapsack problem consists of NP-hard features including the need of combinatorial optimization [37]. Due to its flexible scalability in terms of the domain size, the multi-objective knapsack problem is a preferable choice of experiment for our proposed model, since it is specialized in solving problems with varying degrees of dimension. We tested each algorithm on multi-objective optimization knapsack benchmark problem sets [38].

##### 1) EXPERIMENT SETTINGS

Our experiment consists of 10 different knapsack problems with three objective functions, and we experimented 30 times on each problem, increasing the dimension from 10 to 100 dimensions. We initialized the parameters with uniform random numbers, and the EDA parameter setting was performed 10 times under the same conditions and then averaged. Therefore, the simulations were conducted under the same situations, thus guaranteeing a fair comparison.

In order to optimize the Pareto set, we build a promising candidate solution set by borrowing non-dominated sorting, crowding distance sorting, a representative NSGA-II [14] technique of Pareto-dominated MOEA. For this reason, we used NSGA-II as a comparative model of MODEDA that we proposed.

We also replaced the modeling and sampling procedure of the MOEA with a representative variable auto-encoder (VAE) [34] to effectively solve the multi-dimensional optimization problem LSMOP, sampling a set similar to the probability distribution of population latent factors, and improving the scalability of the model based on dimensionality. RBM-EDA [27], DBN-EDA [28] builds probability models that estimate the distribution of promising candidate solutions, and samples them to create new candidate solutions. For this reason, we selected RBM-EDA and DBN-EDA and compared the performance of our proposed MODEDA.

##### 2) PARAMETER SETTINGS

###### a: REPRODUCTION OPERATORS

The population and iteration size is set to 1000 and 50 equally in the NSGA-II, MODEDA, RBM-EDA, and DBN-EDA algorithms. The crossover and mutation rates of NSGA-II, MODEDA, RBM-EDA, and DBN-EDA are randomly picked up from [0.7, 0.9] and from  $[1/n, 2/n]$  and the binary tournament selection scheme is used. The  $n$  is defined as the length of the gene that is problem size.

###### b: DEEP NETWORKS SETTING

The number of visible neurons of MODEDA, RBM-EDA, and DBN-EDA is set to problem size. The number of hidden neurons in RBM-EDA is set to  $n \times 0.5$ , and DBN-EDA and MODEDA added another hidden layer of  $n \times 0.8$  in between the visible layer and top hidden layers. The hidden layer size and number of latent variables were set to 512 and 64, respectively. The batch size and learning rate of RBM-EDA,

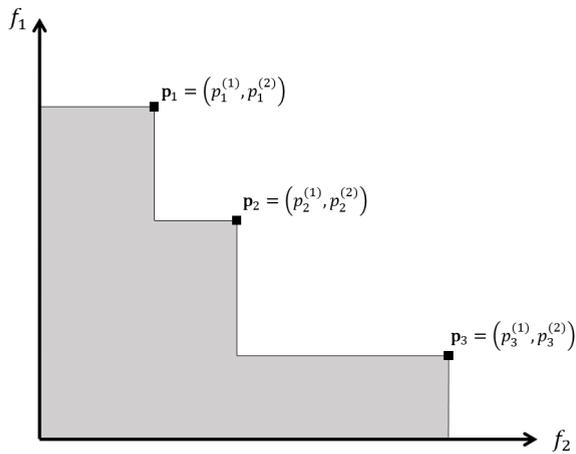


FIGURE 3. A region which is dominated by non-dominated objective vectors  $p_1$ ,  $p_2$ , and  $p_3$ .

and DBN-EDA are set to 32, 0.001, respectively, and parameter initialization is set to uniform random. The activation function of RBM-EDA and DBN-EDA used sigmoid, and the Adam optimizer [39] is set to  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . In our proposed MODEDA, where the batch size and learning rate are set to 32 and 0.001, the activation function used ReLU, the Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  is used to train our VAE and latent space size is used as  $n \times 0.25$ .

c: SPECIFIC PARAMETER SETTINGS IN EACH ALGORITHM

MODEDA, RBM-EDA and DBN-EDA are all used as  $K$ -means/ $K = 3$  for  $K$ -means/ $K = 3$ . In RBM-EDA and DBN-EDA, the contrastive divergence step is set to 1 for training the RBM and DBN and the Gibbs sampling step is used to 25 for sampling new independents.

3) PERFORMANCE INDICATORS

We used the hypervolume indicator [40] to measure the quality of the non-dominated solutions which are obtained by each algorithm. The indicator calculates region which is dominated by the set of non-dominated objective vectors. When the number of objective is  $n$ , it is represented as  $p = \{p^{(1)}, p^{(2)}, \dots, p^{(n)}\}$ . Figure 3 shows an example of hypervolume when maximizing two conflicting objectives  $f_1$   $f_2$ . The filled region means the dominated filled by non-dominated objective vectors which are  $p_1$ ,  $p_2$  and  $p_3$  when the reference point is  $(0, 0)$  and the reference point is  $(0, 0)$ . The area of the filled region is a hypervolume.

4) EXPERIMENT RESULTS

Figure 4 shows the degree of performance improvement compared to the algorithm with the lowest performance. Although MODEDA shows the lowest performance in a 20-dimensional problem, it is still greater than the others. We can see that RBM and DBN also show growing performance according to the problem size. However, DBN, which extends the RBM, showed almost the similar performance.

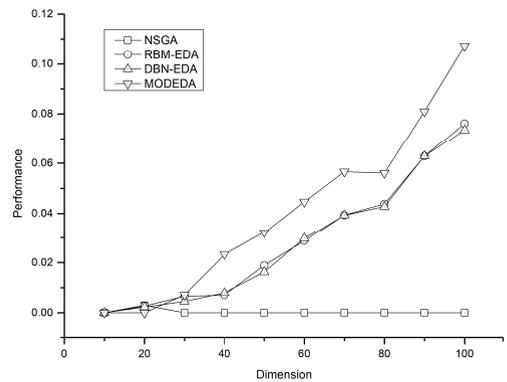


FIGURE 4. Comparison of performance between NSGA-II, RBM-EDA, DBN-EDA and MODEDA according to problem size.

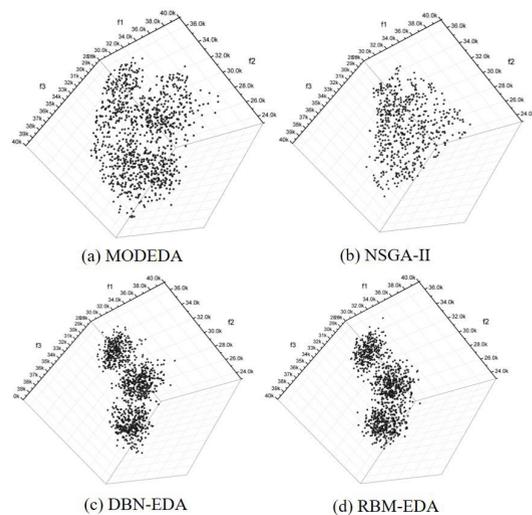


FIGURE 5. Pareto front of MODEDA, NSGA-II, DBN-EDA, RBM-EDA at 50th generation of 100-dimension problem.

RBM-EDA and DBN-EDA did not show a significant difference in terms of optimal performance.

The solution distributions at the last generation of MODEDA, NSGA-II, DBN-EDA, and RBM-EDA in 100-dimension problems are shown in Figure 5. The final search results of NSGA-II are evenly spread in the fitness space, but the quality of the solution was not good compared with the other algorithms. RBM-EDA and DBN-EDA found a higher quality solution set than NSGA-II, but the solutions were biased at the center of each model. MODEDA found a set of solutions with a hypervolume value larger than any other comparative algorithm, while the solutions are distributed evenly in the fitness space, like the NSGA-II.

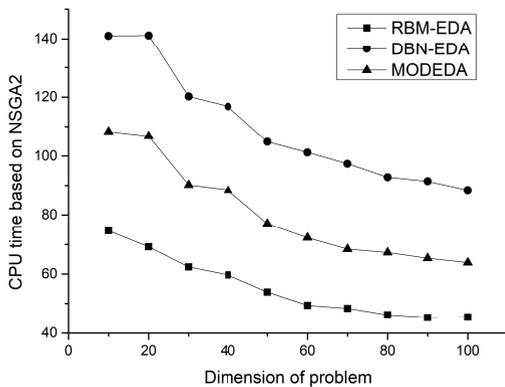
Table II shows the mean of the hypervolume and standard deviation values performed 30 times for each problem of NSGA-II, RBM-EDA, DBN-EDA, and MODEDA. MODEDA showed the highest hypervolume value for benchmark problems which are higher than 30 dimensions, as shown in Table II. However, in the relatively small 20-dimensional, RBM performed the best.

**TABLE 2.** Comparison of hypervolume and standard deviation of the 10 test problems which are growing dimension.

| n   | NSGA-II            | RBM-EDA             | DBN-EDA                    | MODEDA                     |
|-----|--------------------|---------------------|----------------------------|----------------------------|
| 10  | <b>5.53E+10(0)</b> | 5.527E+10(0)        | 5.527E+10(0)               | 5.527E+10(0)               |
| 20  | 3.62E+11(2.84E+08) | 3.617E+11(4.73E+08) | <b>3.616E+11(6.39E+08)</b> | 3.608E+11(9.50E+08)        |
| 30  | 1.55E+12(8.06E+09) | 1.563E+12(9.24E+09) | 1.560E+12(1.02E+10)        | <b>1.564E+12(3.74E+09)</b> |
| 40  | 3.16E+12(2.42E+10) | 3.180E+12(1.74E+10) | 3.183E+12(2.45E+10)        | <b>3.231E+12(2.32E+10)</b> |
| 50  | 5.13E+12(5.47E+10) | 5.225E+12(2.67E+10) | 5.212E+12(2.42E+10)        | <b>5.294E+12(2.2E+10)</b>  |
| 60  | 1.44E+13(1.25E+11) | 1.480E+13(6.31E+10) | 1.481E+13(4.85E+10)        | <b>1.503E+13(4.4E+10)</b>  |
| 70  | 2.18E+13(2.36E+11) | 2.267E+13(1.64E+11) | 2.266E+13(9.4E+10)         | <b>2.304E+13(7.29E+10)</b> |
| 80  | 3.24E+13(4.16E+11) | 3.387E+13(1.32E+11) | 3.383E+13(1.49E+11)        | <b>3.427E+13(1.43E+11)</b> |
| 90  | 4.33E+13(5.82E+11) | 4.60E+13(1.90E+11)  | 4.60E+13(3.35E+11)         | <b>4.68E+13(1.63E+11)</b>  |
| 100 | 4.85E+13(8.43E+11) | 5.220E+13(2.43E+11) | 5.207E+13(2.02E+11)        | <b>5.371E+13(3.70E+11)</b> |

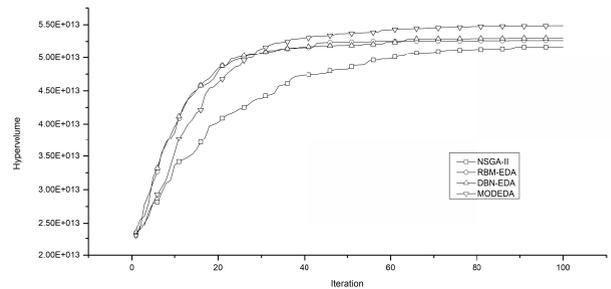
**TABLE 3.** Comparison of hypervolume and standard deviation of the 10 different 100-dimensional multi-objective knapsack problems.

| P  | RBM-EDA                      | DBN-EDA                      | MODEDA                        |
|----|------------------------------|------------------------------|-------------------------------|
| 1  | 5.220E+13(2.43E+11)          | 5.207E+13( <b>2.02E+11</b> ) | <b>(5.371E+13)</b> (3.70E+11) |
| 2  | 6.437E+13(3.88E+11)          | 6.437E+13( <b>1.33E+11</b> ) | <b>(6.557E+13)</b> (1.80E+11) |
| 3  | 4.277E+13(3.80E+11)          | 4.295E+13(5.97E+11)          | <b>4.428E+13(2.41E+11)</b>    |
| 4  | 5.787E+13(6.62E+11)          | 5.660E+13( <b>2.61E+11</b> ) | <b>(5.939E+13)</b> (3.06E+11) |
| 5  | 6.259E+13(4.48E+11)          | 6.273E+13(4.87E+10)          | <b>6.392E+13(3.27E+11)</b>    |
| 6  | 6.468E+13( <b>1.62E+11</b> ) | 6.500E+13(5.14E+11)          | <b>6.633E+13(3.59E+11)</b>    |
| 7  | 7.176E+13(2.46E+11)          | 7.183E+13(1.38E+11)          | <b>7.268E+13(9.49E+10)</b>    |
| 8  | 6.853E+13(6.76E+11)          | 6.875E+13(6.39E+11)          | <b>7.032E+13(2.17E+11)</b>    |
| 9  | 6.250E+13(3.72E+11)          | 6.257E+13(3.96E+11)          | <b>6.384E+13(3.59E+11)</b>    |
| 10 | 5.929E+13(1.62E+11)          | 5.923E+13( <b>1.43E+11</b> ) | <b>6.024E+13(2.79E+11)</b>    |



**FIGURE 6.** Relative computation time compared to NSGA of RBM-EDA, DBN-EDA and MODEDA for increasing problem size.

We also conducted an experiment on 10 different multi-objective knapsack problems, whose dimension and number of objectives are 100 and 3. The values in Table III are the hypervolume and the standard deviation values, which are obtained by 10 repetitions under the same conditions. As the results in Table III show, the proposed MODEDA showed the highest optimization performance for all benchmark problems. RBM-EDA and DBN-EDA did not show a significant difference in terms of optimization performance.



**FIGURE 7.** Hypervolume convergence of NSGA-II, RBM-EDA, DBN-EDA and MODEDA in 3 objectives and 100 size knapsack problem.

Figure 6 shows the relative computation time of RBM-EDA, DBN-EDA, and MODEDA compared to NSGA-II. In cases of small-sized problems, RBM-EDA, DBN-EDA, and MODEDA were very inefficient, requiring about 100 times more computational time than NSGA-II. However, as the size of the problem increases, the relative computation time decreases.

Figure 7 shows a convergence graph of the hypervolume value according to the number of generations of the NSGA-II, RBM-EDA, DBN-EDA, and MODEDA. In this result, NSGA-II could not reach the hypervolume value, which MODEDA obtained at the 50th generation, even after

100 generations. However, after the 1000th generation, NSGA-II could obtain comparative hypervolume to the MODEDA.

## B. DISCUSSION

We proposed a new Generative Deep Network-based EDA for multi-objective optimization and measured its performance on the various sizes of multi-objective knapsack problems. We conducted an experiment with 10 to 100 dim dimensions for 10 knapsack problems and compared the performance of our models with NSGA-II, RBM-EDA, and DBN-EDA. For low-dimensional multi-objective problems (MOPs) between 10 and 20, the proposed algorithm showed comparable or inferior performance to comparison group algorithms (NSGA-II, RBM-EDA, DBN-EDA). For problems that can be modeled with small or simple models, deep networks have weaknesses due to overfitting problems.

As a result, the NSGA-II model showed slightly better performance than the RBM-EDA, DBN-EDA, and MODEDA applied with deep networks in 10 to 20 dimensions. On the other hand, as experimental results show, as problems become more complex and larger, models with Deep Network-based models are more likely to model and sample optimized Pareto set than MOEA models. As shown in Table II and III, compared to deep network models DBN-EDA and RBM-EDA, our model applied with dimension reduction showed competitive performance in LSMOP by showing that the MODEDA model optimized the Pareto set with high hypervolume.

In Figure 5, we compared the range and density of the entire Pareto solution set to see the optimal performance of each model. The RBM-EDA and DBN-EDA applied with the learned deep network showed lower performance in terms of solution distribution, although the higher quality solution of the solution was obtained compared to the existing MOEA-based NSGA-II. However, our MODEDA model demonstrated superiority in optimization performance of the solution set by showing higher performance in both distribution and density of the solution than other models.

The runtime comparison is also shown in Figure 6. By effectively utilizing GPU, which is difficult to utilize by general evolutionary algorithms, our model demonstrates a 70 times reduction in computational time compared to the existing MOEA-based NSGA-II, demonstrating that our model is superior in computational cost.

For the 10 knapsack problems, the convergence profile is shown in Figure 7. We observed that the MODEDA model converges to a higher hypervolume value with lower iteration than the NSGA-II model. Under the same conditions, NSGA-II had to use 20 times more generations to achieve the same results compared to the proposed algorithm. Although the convergence value was slightly lower than that of RBM-EDA and DBN-EDA models, our model finally converged to a higher hyper-volume value, demonstrating superiority in terms of convergence of our model compared to deep network models.

## C. MELODY COMPOSITION EXPERIMENT

### 1) EXPERIMENT SETTINGS

In this paper, two experiments were conducted to evaluate the performance of the proposed algorithm (MODEDA), the Knapsack problem and the music generation. We demonstrated the scalability of the model in the LSMOP problem by sampling a set of solutions similar to the probability distribution of the population through VAE-based dimension reduction by comparing it with RBM-EDA and DBN-EDA through the knapsack experiment.

On the other hand, the music experiment focused on how wide and dense the Pareto set is optimized for the two objective functions of music stability and tension. For this reason, we compared the performance of MODEDA, which was created by borrowing NSGA-II's technique to build a promising candidate solution set, only by comparing it with the Pareto-rule-based NSGA-II algorithm.

### 2) PARAMETER SETTINGS

#### a: REPRODUCTION OPERATORS

The population and iteration size are set to 1000 and 50 both in the NSGA-II and MODEDA. The crossover rate of NSGA-II and MODEDA is randomly selected at [0.7, 0.9], and the binary tournament selection scheme is used. The mutation ratio is set to  $[1/n, 2/n]$ . The  $n$  is defined as the number of notes. One-point crossover is used as the crossover type, and bitflip is used as the mutation type. The parameter initialization of MODEDA is used as uniform random.

#### b: DEEP NETWORKS SETTING

For the deep network-based MODEDA, the experiment was further constructed under the following conditions. The number of hidden neurons in MODEDA is set to  $n \times 0.5$ , and added another hidden layer of  $n \times 0.8$  in between the visible layer and top hidden layers. The hidden layer size and number of latent variables were set to 512 and 64, respectively. The batch size and learning rate are set to 32, 0.001, respectively, the activation function used ReLU, the Adam optimizer [39] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  is used to train our VAE and latent space size were used as  $n \times 0.25$ . The setting for clustering of MODEDA is  $K$ -means/ $K = 3$ .

### 3) CONSTRAINTS

Our method of expressing the melody is a one-hot vector. For melodic representations, mapping pitch to integer values creates many problems in terms of harmony. Thus, we transformed the integer pitch value into a one-hot vector to train the generative network. The total length of the gene is expressed as  $(\max \text{pitch} - \min \text{pitch} + 1) \times (\text{number of notes})$ , and the determinant is an integer between [60, 76], an integer representing the lowest note  $C3$  of the melody and the highest note  $E4$ . To replace the mutations used in NSGA-II, MODEDA applied a uniform random integer from an integer between  $[\max \text{pitch}, \min \text{pitch}]$  to an integer between [60, 76].

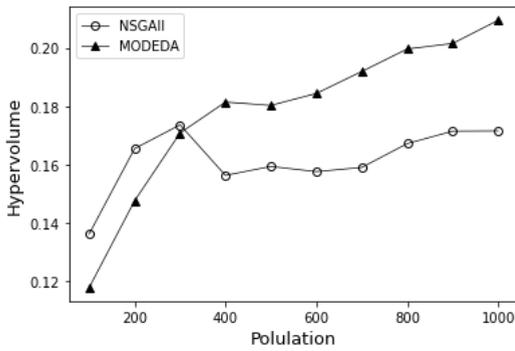


FIGURE 8. Hypervolume according to the size of the population of the NSGA-II and MODEDA.

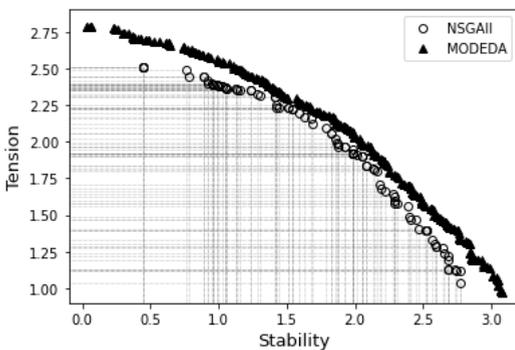


FIGURE 9. Pareto front of the NSGA-II and MODEDA.

4) EXPERIMENT RESULTS AND DISCUSSION

Figure 8 shows the graph of the hypervolume value according to the size of the population of NSGA-II and MODEDA. The graph suggests that our proposed algorithm showed lower performance than NSGA-II between population size 100 and 300. In the case of problems that can be modeled in small size or simple models, deep networks have weaknesses due to overfitting problems. On the other hand, the experiment result shows that Deep Network-based modeling was more capable of modeling and sampling the multi-objective optimization problem as the population size gets larger than 400.

This means that MODEDA found a set of solutions that have larger hypervolume values than NSGA-II as the problems became large. In other words, MODEDA is more powerful to model large problems than the conventional multi-objective algorithms such as NSGA-II, and the continuity of solutions in the latent space enables the proposed algorithm to discover new non-dominated solutions effectively.

Figure 9 illustrates a set of two non-dominated solutions obtained by NSGA-II and MODEDA, respectively. As shown in Figure 9, the obtained Pareto front of our proposed algorithm was a well-spread and dense set of solutions through convex lines compared to the Pareto front of NSGA-II. It denotes that MODEDA has a broad and dense non-dominant solution in two aspects, stability and tension, thereby verifying the excellence of our model.



FIGURE 10. Composition results of MODEDA which are picked at two extreme cases at the 1000 population with 50 generation.



FIGURE 11. Composition results of NSGA-II which are picked at two extreme cases at the 1000 population with 50 generation.

Furthermore, we analyzed melodies picked from the Pareto front of NSGA-II and MODEDA, respectively. A given chord progression was “*CM7 - Am7 - Dm7 - G7*”. The given rhythm consisted of a simple pattern of 4/4 beat, with all four bars being quarter notes.

Figure 10 represents a solution (melody) of MODEDA with the best fitness value in terms of stability and tension. First, when analyzing a stable melody, every first note of each measures (notes of the strong beats) was a chord tone. It consists of a passing tone at the weak beat of the first bar that walks stepwise between notes in two successive chords in the same direction. The Neighboring tone on the weak beat at the fourth bar is a step above on opposite direction. Most of the melodies continued smoothly and consisted of descending progressions, making the melody significantly melodious and stable to listen.

For analyzing a tension melody, it consists of skip or leaps motion that represents large intervals between the adjacent notes. The first notes of the first and second measures, which are the positions of strong beat, consisted of non-chord tones and were solved by the subsequent chord tones. In addition, Escape tone, Cambiata, and passing tone were used on the weak beat. The passing tone on the strong beat is composed of an Appgiatura, which changes the song and adds tension.

Figure 11 represents a solution (melody) of NSGA-II with the best fitness value in terms of stability and tension. As shown in Figure 11, the composition results of NSGA-II exhibited some differences in each melody in terms of stability and tension. The composition results from NSGA-II, there are many elements that deviate from both tension and stability: unresolved tension, avoid on strong beats, excessive leap motions. Moreover, the composition experiment shows that NSGA-II was less optimized than MODEDA at the same 50th generation. It denotes that NSGA-II was less optimized than MODEDA at the same 1000 population. Under the same population conditions, NSGA-II requires more generations to obtain optimization results compared to the proposed algorithm. Our proposed model is superior in rapidly converging to a higher hypervolume in terms of the number of generations.

In the second composition, we conducted a composition experiment with twice the number of measures and various

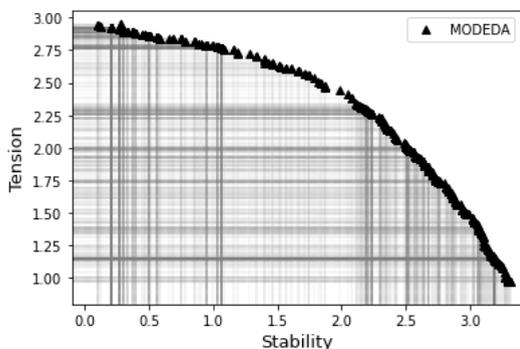


FIGURE 12. Pareto front of the MODEDA at Second composition test.



FIGURE 13. The Solution that has the best fitness value in terms of the stability fitness function.



FIGURE 14. The Solution of the between two extreme solutions.

rhythm patterns to solve more high-dimensional and complex problems. The composition results are obtained by the proposed method. A given chord progression was “C - G - Am - Em - F - C - Dm7 - G”. The given rhythm pattern was borrowed from Canon by the German Baroque composer Johann Pachelbel. Figure 12 shows that even in complex and higher-dimensional problems, MODEDA can obtain a wide and dense Pareto solution set.

Figure 13 shows the melody with the best fitness in terms of the stability fitness function. Most of the first notes in each measure consist of a chord tone, and the weak beats gently lead to non-chord-tone passing tone and escape tone. Looking at the overall composition of the melody, the range of the lowest and highest notes is small. Moreover, the melody consists of conjunct motion or step motion which is the difference in pitch between two consecutive notes of a musical scale. Finally, the melody is solved with a note that descends sequentially, giving a sense of stability to the entire atmosphere of the song.

Figure 14 is the middle point of the pareto front of Figure 12 to extract a melody in which two elements of stability and tension are mixed. There were non-chord tone notes that created tension on the strong beat at the second, fourth, and seventh bars. The melody contained Escape tones



FIGURE 15. The Solution that has the best fitness value in terms of the tension fitness function.

and Passing tones. It also consists of a melody that shows frequently leaps and descends. We confirmed that the melody in Figure 14 has more tension elements compared to the melody in Figure 13.

Figure 15 shows the melody with the best fitness in terms of the tension fitness function. For the second bar of the melody, arranging step motions that minimize the movements of the notes in the beginning and leap motions with loud notes in the latter part added heightened atmospheres of the song. Looking at the overall composition of the melody, within the large range, the sound moves up and down frequently. In addition, the frequency of leaps is high, making it sound tense and elevated overall.

In summary, we verified the multi-objective dimensional performance of MODEDA through a four-bar melody consisting only of a quarter note and an eight-bar melody with a more complex rhythm. For readers who want to listen to the music composed by our proposed model, the sheet music was uploaded to Soundcloud website (<https://soundcloud.com/emptymoon0115>) as a sound source.

## V. CONCLUSION AND FUTURE RESEARCH

In this paper, we propose a new multi-objective generative deep network-based distribution algorithm (MODEDA) to effectively solve Large-Scale Multi-Objective Optimization problems (LSMOP) such as music composition. Due to the VAE’s ability to learn and create search space dimensions, MODEDA improves the scalability of LSMOP, which is effective in solving problems. About the difficulties with dimensional transformation [29], MODEDA proposes a new method that searches for new solutions in a low-dimensional converted decision space and maps them to the solution in the original decision space to ensure consistency with the pareto front solution set.

Through various size knapsack experiments and performance comparison with NSGA-II, RBM-EDA, and DBN-EDA, the proposed model was optimized for relatively high-dimensional problems and proved to be excellent. In addition, through music experiments, we verified the excellence of the proposed model performance by demonstrating that it has a more dense and extensive Pareto-front set than the NSGA-II model and produces a well-converged solution for two musical objective functions: tension and stability.

From the practical point of view of this study, a system that can provide professional composers with different styles of melodies between stability and tension in the same chord progression can be an inspirational and efficient facilitator of the composition process. It will also provide new creative experiences to the general public as a tool to complete melody composition without musical prior knowledge and talent.

We can take some possible directions for future research. When training the proposed VAE, the framework for optimizing the hyper-parameters of each generation for higher performance is open to improvement. In addition, more efforts need to be made to introduce more efficient deep network generation models other than VAE. We are inclined to studying fitness features for structural aspects of melodic composition that are not sufficiently addressed in this study, leaving the expansion to MOPs with three or more objectives to be a future study.

## ACKNOWLEDGMENT

(Jae-Hun Jeong and Eunbin Lee contributed equally to this work.)

## REFERENCES

- [1] A. Gartland-Jones and P. Copley, "The suitability of genetic algorithms for musical composition," *Contemp. Music Rev.*, vol. 22, no. 3, pp. 43–55, Sep. 2003.
- [2] D. Matic, "A genetic algorithm for composing music," *Yugoslav J. Oper. Res.*, vol. 20, no. 1, pp. 157–177, 2010.
- [3] J. A. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proc. ICMC*, vol. 94, 1994, pp. 131–137.
- [4] B. Johanson and R. Poli, "GP-music: An interactive genetic programming system for music generation with automated fitness raters," in *Proc. Cogn. Sci. Res. Centre*, Jul. 1998, pp. 181–186.
- [5] M. Dostál, "Musically meaningful fitness and mutation for autonomous evolution of rhythm accompaniment," *Soft Comput.*, vol. 16, no. 12, pp. 2009–2026, Dec. 2012.
- [6] A. Freitas and F. Guimaraes, "Melody harmonization in evolutionary music using multiobjective genetic algorithms," in *Proc. Sound Music Comput. Conf.*, 2011, pp. 1–8.
- [7] M. Scirea, J. Togelius, P. Eklund, and S. Risi, "Metacompose: A compositional evolutionary music composer," in *Proc. Int. Conf. Evol. Biol. Inspired Music Art*, 2016, pp. 202–217.
- [8] J. Jeong, Y. Kim, and C. W. Ahn, "A multi-objective evolutionary approach to automatic melody generation," *Expert Syst. Appl.*, vol. 90, pp. 50–61, Dec. 2017.
- [9] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.
- [10] Z. P. Stanko, T. Nishikawa, and S. R. Paulinski, "Large-scale multi-objective optimization for the management of seawater intrusion, Santa Barbara, CA," in *Proc. AGU Fall Meeting*, 2015.
- [11] Z. Zhao, M. Jiang, S. Guo, Z. Wang, F. Chao, and K. C. Tan, "Improving deep learning based optical character recognition via neural architecture search," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–7.
- [12] J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. Garcia-Nieto, F. Luna, and E. Alba, "A study of multi objective meta heuristics when solving parameter scalable problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 618–635, Feb. 2010.
- [13] E. T. Oldewage, A. P. Engelbrecht, and C. W. Cleghorn, "The merits of velocity clamping particle swarm optimisation in high dimensional spaces," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [15] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.
- [16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [17] N. Siddique and H. Adeli, "Computational intelligence: Synergies of fuzzy logic," in *Neural Networks and Evolutionary Computing*. Hoboken, NJ, USA: Wiley, 2013.
- [18] K. Deb, "Multi-objective optimization," *Search Methodologie*. Boston, MA, USA: Springer, 2014, pp. 403–449.
- [19] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization," *Complex Intell. Syst.*, vol. 3, pp. 247–263, Dec. 2017.
- [20] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Mar. 2015.
- [21] G. Kirlik and S. Sayin, "A new algorithm for generating all non-dominated solutions of multi objective discrete optimization problems," *Eur. J. Oper. Res.*, vol. 232, no. 3, pp. 479–488, 2014.
- [22] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, Jun. 2003.
- [23] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [24] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multi objective optimization," in *Proc. 3rd Annu. Conf. Genetic Evol. Comput.*, 2001, pp. 283–290.
- [25] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multi objective optimization," in *Proc. Evol. Methods Design, Optim., Control*, Jul. 2002, pp. 95–100.
- [26] P. Larranaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, vol. 2. New York, NY, USA: Springer, 2001.
- [27] M. Probst, F. Rothlauf, and J. Grahl, "Scalability of using restricted Boltzmann machines for combinatorial optimization," *Eur. J. Oper. Res.*, vol. 256, no. 2, pp. 368–383, Jan. 2017.
- [28] S. Saikia, L. Vig, A. Srinivasan, G. Shroff, P. Agarwal, and R. Rawat, "Neuro-symbolic EDA-based optimization using ILP-enhanced DBNs," in *Proc. Integrating Neural Symbolic Approaches Co-Located 30th Annu. Conf. neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 1–9.
- [29] W.-J. Hong, P. Yang, and K. Tang, "Evolutionary computation for large-scale multi-objective optimization: A decade of progresses," *Int. J. Autom. Comput.*, vol. 18, no. 2, pp. 155–169, Apr. 2021, doi: 10.1007/s11633-020-1253-0.
- [30] H. Wang, L. Jiao, R. Shang, S. He, and F. Liu, "A memetic optimization strategy based on dimension reduction in decision space," *Evol. Comput.*, vol. 23, no. 1, pp. 69–100, Mar. 2015.
- [31] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3129–3142, Jun. 2021.
- [32] Z. Wang, H. Hong, K. Ye, G.-E. Zhang, M. Jiang, and K. C. Tan, "Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 29, 2021, doi: 10.1109/TNNLS.2021.3113158.
- [33] H. Hong, K. Ye, M. Jiang, D. Cao, and K. C. Tan, "Solving large-scale multiobjective optimization via the probabilistic prediction model," *Memetic Comput.*, vol. 14, no. 2, pp. 165–177, Jun. 2022, doi: 10.1007/s12293-022-00358-9.
- [34] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
- [35] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014, *arXiv:1401.4082*.
- [36] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective hBOA, clustering, and scalability," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2005, pp. 663–670.
- [37] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2007, pp. 862–876.
- [38] C. Roig, L. J. Tardón, I. Barbancho, and A. M. Barbancho, "Automatic melody composition based on a probabilistic model of music style and harmonic rules," *Knowl.-Based Syst.*, vol. 71, pp. 419–434, Nov. 2014.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.  
 [40] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2007, pp. 862–876.



**JAE-HUN JEONG** received the B.S. degree in computer science from Sungkyunkwan University, South Korea, in 2014, and the M.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Sungkyunkwan University, in 2018. He is currently working as a CTO with Creativemind Inc., South Korea. He developed an artificial intelligent music composer 'EVOM' that collaborated with various artists and researchers. His research interests include machine learning techniques for artificial music composition, especially evolutionary computing and generative neural networks.



**EUNBIN LEE** received the B.S. degree in physics and materials science and engineering from Yonsei University, Wonju, South Korea, in 2020. She is currently pursuing the M.S. degree with the School of Artificial Intelligence, Gwangju Institute of Science and Technology (GIST). She is working on composition using artificial intelligence under the stage name of Emptymoon0115. Her research interests include generative neural networks and evolutionary computation for music composition and singing voice synthesis.



**JONG-HYUN LEE** received the B.S., M.S., and Ph.D. degrees in computer science from Sungkyunkwan University, in 2010, 2012, and 2017, respectively. He has been working as a Postdoctoral Researcher with the IT Convergence Research Institute, Sungkyunkwan University, since 2017, and as a CEO at Creativemind Inc. His research interests include artificial intelligence, machine learning, evolutionary algorithm, and swarm robotics.



**CHANG WOOK AHN** (Member, IEEE) received the Ph.D. degree from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2005. From 2005 to 2007, he worked with the Samsung Advanced Institute of Technology, South Korea. From 2007 to 2008, he was a Research Professor at the GIST. From 2008 to 2016, he was an Assistant/Associate Professor at the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. He is currently working as a Professor with the School of Artificial Intelligence, GIST. His research interests include genetic algorithms/programming, multi-objective optimization, neural networks, and quantum machine learning.

•••