Combining Artificial Neural Networks and Evolution to Solve Multiobjective Knapsack Problems

Roman Denysiuk
CEA LIST
Gif-sur-Yvette, France
roman.denysiuk@cea.fr

António Gaspar-Cunha
University of Minho
Guimarães, Portugal
agc@dep.uminho.pt

Alexandre C. B. Delbem
University of São Paulo
São Carlos, Brazil
acbd@icmc.usp.br

ABSTRACT

The multiobjective knapsack problem (MOKP) is a combinatorial problem that arises in various applications, including resource allocation, computer science and finance. Evolutionary multiobjective optimization algorithms (EMOAs) can be effective in solving MOKPs. Though, they often face difficulties due to the loss of solution diversity and poor scalability. To address those issues, our study [2] proposes to generate candidate solutions by artificial neural networks. This is intended to provide intelligence to the search. As gradient-based learning cannot be used when target values are unknown, neuroevolution is adapted to adjust the neural network parameters. The proposal is implemented within a state-of-the-art EMOA and benchmarked against traditional search operators based on binary crossovers. The obtained experimental results indicate a superior performance of the proposed approach. Furthermore, it is advantageous in terms of scalability and can be readily incorporated into different EMOAs.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**; **Search methodologies**;

KEYWORDS

Evolutionary computing, artificial neural networks, multiobjective knapsack problem

ACM Reference Format:
Roman Denysiuk, António Gaspar-Cunha, and Alexandre C. B. Delbem. 2019. Combining Artificial Neural Networks and Evolution to Solve Multiobjective Knapsack Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The knapsack problem is a well-known combinatorial optimization problem. It comprises a knapsack of certain capacity and a set of items characterized by weights and profits. The aim is to fill up the knapsack by a collection of items so that the total profit is maximized and the total weight does not exceed the given capacity. There are several variants of a knapsack problem. The most common one is the 0-1 knapsack problem, where the number of copies of each item is either zero or one. MOKP is an extension of a single objective knapsack problem [7] and can be formulated as follows.

$$\begin{aligned} \underset{\boldsymbol{x} \in \{0,1\}^n}{\text{maximize}} \quad & f_j(\boldsymbol{x}) = \sum_{i=1}^{n} x_i p_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i w_{ij} \leq c_j \qquad \forall j \in \{1, \ldots, m\} \end{aligned}$$

where m is the number of knapsacks, n is the number of items, p_{ij} and w_{ij} are respectively the profit and weight of the i-th item with respect to the j-th knapsack, c_j is the capacity of the j-th knapsack.

A feasibility ration r_j determines the capacity of the j-th knapsack as $c_j = r_j \sum_{i=1}^{n} w_{ij}$.

A binary string $\boldsymbol{x}$ encodes the solution such that

$$\forall i \in \{1, \ldots, n\} : \; x_i = \begin{cases} 1 & \text{if the } i\text{-th item is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Due to its practical importance, the knapsack problem has been intensively investigated. There have been developed different exact and approximate algorithms. This study aims to address difficulties experienced by EMOAs on MOKPs, such as the loss of diversity, poor convergence and scalability [4, 6]. The mechanism for generating solutions, aimed to exploit learning capabilities of ANNs, is suggested. The intelligence to this mechanism is provided by using the item profits and weights as input data. This input is propagated through neural networks whose outputs offer decisions on the selection of items in multiobjective search. Although this resembles a traditional classification task, there are no target values for training and gradient-based learning cannot be applied. Therefore, it is suggested to evolve network networks by evolutionary process, which is known as neuroevolution [3].

2 METHOD

The $\mathcal{S}$ metric selection evolutionary multiobjective optimization algorithm (SMS-EMOA) [1] is adopted as the baseline framework for operating in the objective space. SMS-EMOA evolves the population using a steady-state evolutionary process where a single offspring is produced in each generation. The mating selection is performed by picking up uniformly at random a set of parent individuals. The variation procedure generates single offspring manipulating the selected parents. The replacement procedure updates the population by removing an individual with the smallest hypervolume contribution in the last nondominated front.

For generating offspring, the study suggests to use artificial neural networks (ANNs) instead of traditional evolutionary operators. This explores the nature inspired notion of genotype-phenotype mapping [5]. The genotype space defines a search space where

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

evolutionary operators act for reproduction. The phenotype space represents a space of actual solutions to the problem at hand. The phenotype representation is used to compute the objective values for the given solution.

In our study, a solution the genotype space is represented by a chromosome encoding parameters of neural network. We use a direct encoding with one-to-one mapping between genes and network parameters. Each chromosome is represented by a real-valued string. Depending on the location, a gene encodes either weight, bias or a component of boolean mask for the hidden layer. The boolean mask is determined by thresholding at zero. The false values indicate the removal of neurons with all incoming and outgoing connections. This is a simple yet effective encoding scheme that can learn the parameters and topology of neural networks.

A suitable genotype-phenotype mapping is defined as follows. An individual's chromosome is decoded into the feedforward neural network with one hidden layer. The standard sigmoid function is used in all hidden and output neurons. The items associated with the MOKP are placed into a feature space. The simplest way is to use normalized weights and profits of the item as features. The features of each item are fed into the neural network. The output neuron produces binary variables by thresholding indicating the items selection.

To achieve a desired behavior, i.e. the optimal genotype-phenotype mapping, the population of chromosomes undergoes the evolution. The selection and replacement are performed using the framework of SMS-EMOA. As to recombination, a real valued encoding enables the application of established operators for continuous optimization. The effects of different operators are investigated in the experimental study. Thus, our study revolves around the idea of combining neural networks and evolutionary computing to exploit the strengths of both.

3 RESULTS

The proposed approach has been validated through computational experiments. These involved MOKP instances having between 2 and 5 knapsacks with items ranging from 500 to 10000. All the tested algorithms performed 21 independent runs using different seeds for the random number generator. The population size and the maximum number of generations were set to 100 and 1000, respectively. The hypervolume and epsilon indicators were used for performance comparison. The Wilcoxon rank sum test was applied to draw statistically sound conclusions.

The experiments investigated the following issues.

1. The effectiveness in comparison with traditional evolution. We compared evolutionary and neuroevolution search having them operating identically in the objective space. The former evolved binary strings using one-point, two-point and uniform crossovers. The latter evolved neural networks for genotype-phenotype mapping. As a result, neuroevolutionary approach significantly outperformed evolutionary ones regarding both convergence and diversity. The results also revealed neuroevolution performs superior in terms of scalability. This is because the size of the search space explored by neuroevolution depends slightly on the MOKP size.

2. The effect of variation operator. This operator is responsible for generating offspring and is important for the effectiveness of the search process. One advantage of the used real encoding is the ability to adapt different variation operators proved effective in continuous optimization. Three variation operators were investigated, namely simulated binary crossover (SBX), differential evolution (DE) and evolution strategy (ES). Better results were obtained with ES operator. This can be explained by a self-adaptation of strategy parameters, such as the mutation strengths encoded into the chromosome.

3. The effect of feature mapping. Two possible ways to represent items in the feature space were investigated. The one used normalized profits and weights of items. This mapping is simple and leads to the size of input layer being twice the size of the objective space. The other used the ratio between profits and weights. Such mapping can be useful because it reduces the number of input neurons and parameters in neural network. The results showed that the first approach performs better. A possible reason is that the features represented by the ratio between profits and weights are less informative.

4. The effect of network topology. The topology refers to the number of neurons and the way they are connected in ANN. This is an important issue as it influences the learning and determines the expressive capacity. We studied the ability of neuroevolution to determine the optimal network structure. We compared the results obtained with a fixed number of neurons in the hidden layer and those determined by search. Slightly better results were produced using a fixed topology. This can be due to an additional complexity introduced into the search.

4 CONCLUSIONS

This study suggested a neuroevolutionary approach to solve MOKPs. Contrary to traditional evolutionary approaches evolving solutions to the MOKP, the proposed approach evolves genotype-phenotype mappings. The obtained results showed it can produce outperformance and exhibit scalability.

As future work, the introduced idea can be adapted to solve similar combinatorial problems. Another interesting research direction is to explore the notion of transfer learning, where neural networks resulting from addressing one task are applied to solve other problems.

REFERENCES

[1] N. Beume, B. Naujoks, and M. Emmerich. 2007. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *Eur. J. Oper. Res.* 181, 3 (2007), 1653–1669.
[2] R. Denysiuk, A. Gaspar-Cunha, and A. C. B. Delbem. 2019. Neuroevolution for solving multiobjective knapsack problems. *Expert Syst. Appl.* 116 (2019), 65–77. https://doi.org/10.1016/j.eswa.2018.09.004
[3] D. Floreano, P. Dürr, and C. Mattiussi. 2008. Neuroevolution: from architectures to learning. *Evol. Intell.* 1, 1 (2008), 47–62.
[4] H. Ishibuchi, N. Akedo, and Y. Nojima. 2015. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. *IEEE Trans. Evol. Comput.* 19, 2 (2015), 264–283.
[5] F. Rothlauf. 2006. *Representations for Genetic and Evolutionary Algorithms* (2 ed.). Springer, Heidelberg.
[6] H. Sato, H. E. Aguirre, and K. Tanaka. 2007. Local dominance and local recombination in MOEAs on 0/1 multiobjective knapsack problems. *Eur. J. Oper. Res.* 181, 3 (2007), 1708–1723.
[7] E. Zitzler and L. Thiele. 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. Evol. Comput.* 3, 4 (1999), 257–271.