Success Rate Based Scaling Factor Adaptation in Dual-Population Differential Evolution

Vladimir Stanovov*[1] and Eugene Semenkin[1]

[1]Siberian Federal University – Russia

Abstract

In this paper a new strategy is proposed for scaling factor adaptation in DE. It is based on the success rate value, i.e. the number of successful solutions generated within one generation divided by total population size. The proposed method is inspired by the results of the genetic programming, which was previously applied to design parameter adaptation schemes. In this study these ideas are further developed, and applied to a recently proposed L-NTADE algorithm. It is shown that on two sets of benchmark functions, used in Congress on Evolutionary Computation 2017 and 2022, the modified algorithm is capable of achieving much better results, and compete with the best proposed algorithms, while using the same set of parameters.

*Speaker

An improved multi-objective genetic algorithm for the neural architecture search problem

Eloy Bedia-García[1] and Enrique Domínguez[1]

Dept. of Computer Science
University of Malaga
eloybg97@uma.es, enriqued@lcc.uma.es

1 Introduction

In recent years, there is a great interest in automating the process of searching for neural network topology. This problem is called Neural Architecture Search (NAS), which can be seen as a 3-gear mechanism: the search space, the error estimation and the search strategy.

The search space defines what kind of neural architectures can be reached, and it can be divided into three types:

- A search space restricted to sequential models. These models consist of consecutive layers $(L_0.L_1, ..., L_n)$, where layer L_i receives as input the output of layer L_{i-1}
- A search space with more complex architectures where the input of the i-th layer is defined by the function $g_i(L_0, ..., L_{i-1})$. Examples of this search space are the Residual Networks [4] or the DenseNets [5].
- The search space of block-based architectures [12], [11]. These are based on a design pattern that consists of the consecutive repetition of a set of layers called blocks.

To guide the selected strategy throughout the search space, we need a metric to help us. The simplest way is to evaluate the error obtained in the validation set, however, due to the long computation times required, alternative methods are being searched for, such as: reducing the training set [12], reducing the number of epochs [10, 12], using less filters [7, 12] or using lower resolution images [2].

Different kind of search strategies have been proposed in the literature mainly based on reinforcement learning with a reinforce policy [9], a proximal policy optimization [12] or Q-learning [1]. Other strategies based on evolutionary algorithms have been also proposed with different types of genetic operators, such as using selection by tournament [7, 8], eliminating the worst individual in each generation [8], eliminating the oldest individual [7], or using Lamarkian inheritance [3] for the offspring generation.

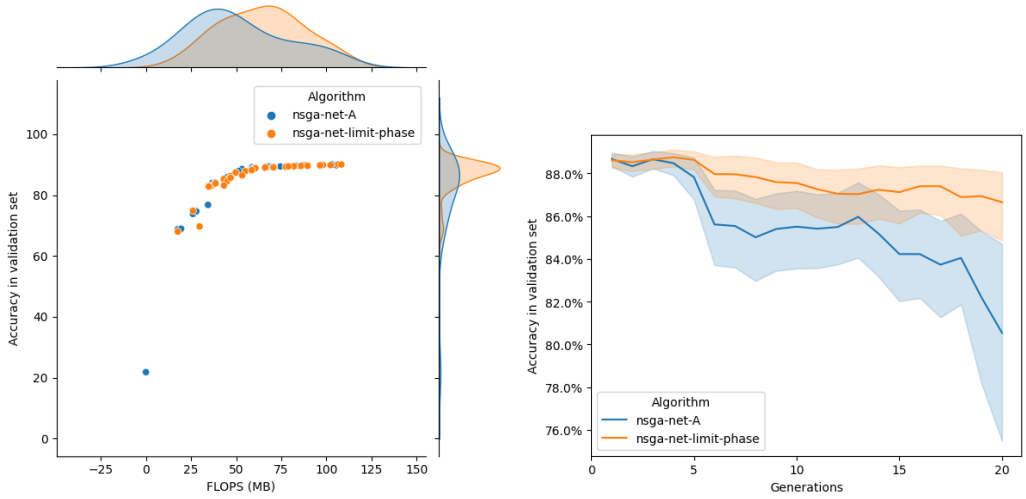
In this paper, we propose an improved version of the NSGA-Net algorithm [6], which is a multi-objective genetic algorithm for the NAS problem. One of the drawbacks is the limited diversity that can be generated by the original crossover operator, which generates only one offspring keeping the common genomes, and leaving the rest randomly. In order to avoid this limitation, we proposed a new 2-point crossover restricting the possible cutoff points only to the block limits.

The rest of the paper is organized as follows. In the next section, some preliminary results are shown on the well-known CIFAR-10 dataset. Finally, some remarks and future works are presented in the conclusions section.

2 Experiments

In this section, we present the empirical results to show the efficacy of the proposed operators for the NSGA-Net algorithm to automate the NAS process on the CIFAR-10 benchmark. Two objectives were considered to guide our NSGA-Net based algorithm: the classification error on the validation set and the computational complexity measured as the number of floating point operations (FLOPs) needed to execute the forward pass of the neural network. The CIFAR-10 dataset was considered for the classification task, splitting the original training set into our training (80%) and validation set (20%) for the neural architecture search. The original testing set was only used to obtain the test accuracy of the final models.

Regarding to the genetic parameters, the population was randomly initialized with size fixed to 40 during the 20 generations for exploration and 10 generations for exploitation. The coding of the genomes consists in a sequence of 3 blocks, each containing a maximum of 4 nodes. The training of the neural networks is carried out for 25 epochs, with a learning rate of 0.025 descending according to the cosine annealing scheme, and a batch size of 128.



(a) Objectives space distribution (b) Evolution of the population accuracy

Fig. 1: NSGA-Net vs LimitPhase on CIFAR-10

Figure 1 shows the comparative result between the original NSGA-Net and our proposed algorithm on the CIFAR-10 dataset. The distribution of the solutions provided by both algorithms is shown in figure 1a, where our proposal (NSGA-LimitPhase) presents a higher accuracy concentration in values around 90%, although with greater complexity (FLOPs). This information is also reflected in figure 1b, where the evolution of the accuracy of the population models of NSGA-LimitPhase presents a better performance. Additionally, in this figure we can see the deterioration of the NSGA-Net population throughout the generations, which it is accentuated at the end. However, our proposal presents accuracies more stable at high values with a smooth downward trend at the end.

3 Conclusions

In this work, we have proposed the use of a new crossover to improve the NSGA-Net algorithm for the NAS problem. Experimental results show a better accuracy distribution in the objectives space, and a better accuracy evolution on the CIFAR-10 dataset. These results are very promising for the treatment of other datasets/benchmarks in order to design a good algorithm for solving the NAS problem. Further works include both the application of the proposed algorithm to other datasets and the comparison to other state-of-the-art algorithms for the NAS problem.

Acknowledgement

The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of a RTX A6000 GPU with 48Gb.

References

1. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning
2. Chrabaszcz, P., Loshchilov, I., Hutter, F.: A downsampled variant of ImageNet as an alternative to the CIFAR datasets https://www.semanticscholar.org/paper/A-Downsampled-Variant-of-ImageNet-as-an-Alternative-Chrabaszcz-Loshchilov/e644a409b4a4c6eaedffe27efbc5c76280b34c61
3. Elsken, T., Metzen, J.H., Hutter, F.: Efficient multi-objective neural architecture search via lamarckian evolution. https://doi.org/10.48550/arXiv.1804.09081, http://arxiv.org/abs/1804.09081
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. https://doi.org/10.1109/CVPR.2016.90, ISSN: 1063-6919
5. Huang, G., Liu, Z., Weinberger, K.: Densely connected convolutional networks p. 12
6. Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., Banzhaf, W.: Nsga-net: Neural architecture search using multi-objective genetic algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 419–427. GECCO '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3321707.3321729, https://doi.org/10.1145/3321707.3321729
7. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Aging evolution for image classifier architecture search. https://www.semanticscholar.org/paper/Aging-Evolution-for-Image-Classifier-Architecture-Real-Aggarwal/7bac3d11824fabe0dc3ac2fee9bfb667e82fba9c
8. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q., Kurakin, A.: Large-scale evolution of image classifiers. https://doi.org/10.48550/arXiv.1703.01041, http://arxiv.org/abs/1703.01041
9. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning **8**(3), 229–256. https://doi.org/10.1007/BF00992696, https://doi.org/10.1007/BF00992696
10. Zela, A., Klein, A., Falkner, S., Hutter, F.: Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. https://doi.org/10.48550/arXiv.1807.06906, http://arxiv.org/abs/1807.06906
11. Zhong, Z., Yan, J., Wu, W., Shao, J., Liu, C.L.: Practical block-wise neural network architecture generation. pp. 2423–2432. https://doi.org/10.1109/CVPR.2018.00257
12. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710. https://doi.org/10.1109/CVPR.2018.00907

Ergodic Annealing: intelligent optimization under uncertainty

Carlo Baldassi, Fabio Maccheroni, Massimo Marinacci, Marco Pirazzini

Artificial Intelligence Lab, Bocconi University

Abstract

NP-hard decision problems in which the cost function is *a priori* unknown to the Decision Maker arise naturally in many applications. We present a novel algorithm, inspired by Simulated Annealing, that exploits the structural randomness of Metropolis exploration in order to simultaneously find the optimal solution and learn its cost. As benchmark cases, we test our algorithm on the Directed Steiner Tree and Traveling Salesman problems. Our results suggest that problems for which Simulated Annealing works with known costs can be efficiently attacked with our algorithm when costs are unknown.

Index Terms

Simulated Annealing, stochastic optimization, reinforcement learning.

This paper is dedicated to the memory of Erio Castagnoli whose first job at the University of Parma was at the "Centro di Calcolo" of the Faculty of Economics.

The authors thank Simone Cerreia-Vioglio, Daniele Durante, Salvatore Greco, and Piero Veronese for very useful discussions as well as PRIN (grant 2017CY2NCA) for financial support. The authors have no conflicts of interest to declare.

1 Introduction

The recent years and events lead to a massive development of content-oriented cloud services. The most popular and voluminous content offered in today's networks are videos that must be efficiently delivered to end customers. The objective of the service provider (root) is to optimize the delivery of content to its costumers (terminals). In this optimization problem the cost is usually assumed to be known (Figure 1, left graph). Yet, in reality it is often unknown because it depends on many stochastic factors, such as the traffic on the network, the level of demand, and so on (Figure 1, right graph).

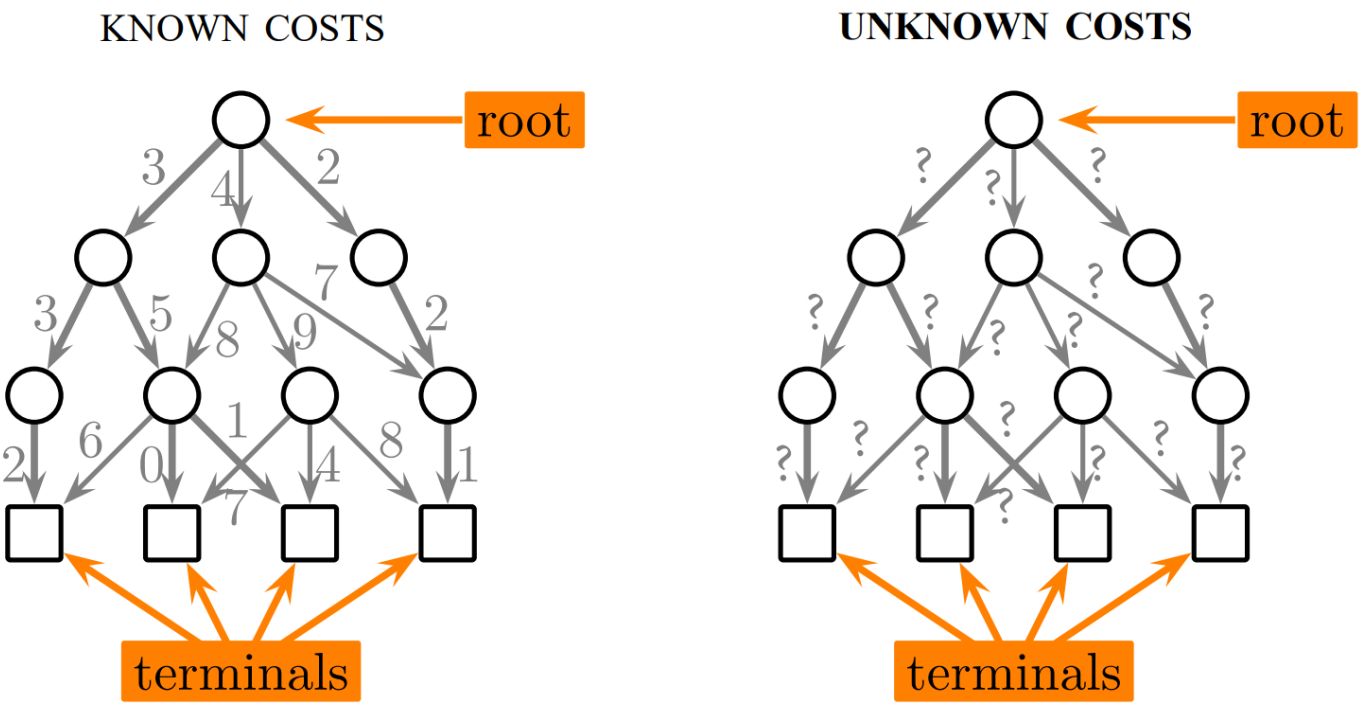


Fig. 1. Directed Steiner trees with known and unknown costs.

This is just an instance of a general decision problem in which, *ex ante*, the Decision Maker (DM) ignores the payoff of the available actions and has limited resources to discover it. In this note, we show that a natural modification of the Simulated Annealing algorithm of Kirkpatrick et al. (1983) permits to efficiently solve this conceptually non-trivial problem.

2 Known payoffs: Metropolis Algorithm and Simulated Annealing

Let A be a finite set of actions and $u : A \to \mathbb{R}$ a **known** objective function that the DM aims to maximize.[1] When the number of alternatives is small, the DM can just use a brute force comparison-and-elimination algorithm that, after $|A| - 1$ binary comparisons, finds the optimal alternative.

When the number of alternatives increases, one needs to go beyond this basic algorithm. In particular, one can rely upon the celebrated *Metropolis Algorithm* (Metropolis et al., 1953) and its evolution called *Simulated Annealing* (Kirkpatrick et al., 1983).

1. It goes without saying that for minimization problems the analysis is similar.

Metropolis Algorithm: Let $\beta > 0$.

Step 0**.** Choose $a \in A$ randomly.

Step $n+1$**.** Choose $b \in A$ randomly.

- If $u(b) \geq u(a)$, then set $a \doteqdot b$.
- If $u(b) < u(a)$, then set $a \doteqdot b$ with probability $e^{\beta[u(b)-u(a)]}$.

Specifically, the random choice at Step 0 is performed by drawing a from an initial distribution $\mu(\cdot)$ on A; the one at Step $n+1$ by drawing b from a Markovian distribution $Q(\cdot \mid a)$ which depends on the incumbent a.[2] The full name of the algorithm is thus *Metropolis Algorithm* with *inverse temperature* β, *initial distribution* μ and *proposal matrix* Q. Its key property is that, if the current state is a, the next state b is determined according to the following transition probabilities

$$P(b \mid a) = \begin{cases} Q(b \mid a) \min\left\{1, e^{\beta[u(b)-u(a)]}\right\} & \text{if } b \neq a \\ 1 - \sum\limits_{c \neq a} Q(c \mid a) \min\left\{1, e^{\beta[u(c)-u(a)]}\right\} & \text{else} \end{cases}$$

Thus, the algorithm realizes an aperiodic and irreducible Markov chain with stationary distribution

$$p_\beta(a) = \frac{e^{\beta u(a)}}{\sum_{b \in A} e^{\beta u(b)}} \qquad \forall a \in A$$

Therefore, the long-run frequency with which a is chosen from A is almost surely $p_\beta(a)$.

The idea of Simulated Annealing is to slowly increase β, while the Metropolis algorithm runs, with the objective of approaching the limit distribution

$$p_\infty(a) = \lim_{\beta \to \infty} p_\beta(a) = \begin{cases} \dfrac{1}{|\arg\max_A u|} & \text{if } u(a) = \max_A u \\ 0 & \text{else} \end{cases}$$

which concentrates on the maximizers of u on A. In the words of its creators: "At each temperature [here $1/\beta$], the simulation must proceed long enough for the system to reach a steady state." Thus, the constant β is a replaced with a sequence β_n of inverse temperatures such that $\beta_n \equiv \beta_{t_0}$ is maintained constant for $n = 0, ..., t_0$, with t_0 large enough to achieve the stable (empirical) frequency

$$\hat{p}_0(a) \approx \frac{e^{\beta_{t_0} u(a)}}{\sum_{b \in A} e^{\beta_{t_0} u(b)}} \qquad \forall a \in A$$

Subsequently, $\beta_n \equiv \beta_{t_1}$ is maintained constant for $n = t_0 + 1, ..., t_1$, with t_1 large enough to achieve the stable (empirical) frequency

$$\hat{p}_1(a) \approx \frac{e^{\beta_{t_1} u(a)}}{\sum_{b \in A} e^{\beta_{t_1} u(b)}} \qquad \forall a \in A$$

and so on, aiming at a long-run frequency

$$\lim_{k \to \infty} \hat{p}_k(a) \approx \lim_{k \to \infty} \frac{e^{\beta_{t_k} u(a)}}{\sum_{b \in A} e^{\beta_{t_k} u(b)}} = p_\infty(a) \qquad \forall a \in A$$

2. The distribution $Q(\cdot \mid a)$ corresponds to the a-th row of a symmetric and irreducible $A \times A$ stochastic matrix Q. This matrix describes the way in which the algorithm explores the landscape A. Irreducibility guarantees full exploration of A, symmetry is intuitive and can be dispensed with. See Hastings (1970) or Madras (2002) for a textbook treatment.

as β_n diverges (so that β_{t_k} diverges too). The sequence $\{\beta_n\}_{n\in\mathbb{N}}$ is called *annealing schedule*.[3]

Simulated Annealing: Let $\beta_n \to \infty$.

Step 0**.** Choose $a \in A$ randomly.

Step $n+1$**.** Choose $b \in A$ randomly.

- If $u(b) \geq u(a)$, then set $a \doteqdot b$.
- If $u(b) < u(a)$, then set $a \doteqdot b$ with probability $e^{\beta_n[u(b)-u(a)]}$.

The algorithm runs until the system "freezes", that is, a stops changing for a fixed number of consecutive iterations (or until a given number N of iterations has been performed). The selected alternative is a candidate maximizer for the objective function u.

The grain of randomness injected in the last line of the pseudo-code allows the DM to escape from local maxima and explore the actions' landscape (especially at small values of β). In the next section, we use this explorative feature of the algorithm to obtain information about u and maximize it, even when u itself is initially unknown.

3 UNKNOWN PAYOFFS: ERGODIC ANNEALING

Now assume that the objective function u is **unknown** to the DM. In particular, consider the important case when $u(a)$ is the expectation of the random payoff $U(a)$ of alternative $a \in A$, which is unknown because the DM ignores the distribution $F_{U(a)}$ of the random variable $U(a)$ itself.

Example *A basic example is when each $U(a)$ has Bernoulli distribution with unknown probability of success $u(a)$.* □

The initial beliefs of the DM about u are represented by a pair $(u_0, C_0) \in \mathbb{R}^A \times \mathbb{N}^A$, where

$$u_0(a)$$

represents the *ex ante* expected payoff assigned to a by the DM and

$$C_0(a)$$

represents the DM's confidence in this evaluation. Intuitively, $C_0(a)$ is the number of "preliminary samples of a" on which the DM bases his expectations.

Example (continued) *Assume there are three alternatives a, b, c. The pair (u_0, C_0) given by*

$$u_0 = (1,0,1) \quad \textit{and} \quad C_0 = (1,1,1)$$

is interpreted as follows: the DM sampled once from each distribution $F_{U(a)}$, $F_{U(b)}$, $F_{U(c)}$ and observed the realizations

$$U(a) = 1 \quad U(b) = 0 \quad U(c) = 1$$

"Sampling" and "observing" can be factual or hypothetical.

3. Originally, $t_k = (k+1)L$ for some fixed "large" loop length $L \in \mathbb{N}$ and $\beta_{t_k} = (1+\rho)^k \beta_0$ for some "small" factor $\rho \in (0, \infty)$.

Analogously, the pair (u_0, C_0) *given by*

$$u_0 = (0.5, 0.7, 1) \quad \textit{and} \quad C_0 = (2, 10, 1)$$

is interpreted as follows: the DM took 2 *samples from* $F_{U(a)}$ *and observed an average value of* 0.5 *(that is, one success and one failure),* 10 *samples from* $F_{U(b)}$ *and observed an average value of* 0.7 *(that is seven successes and three failures), and* 1 *sample from* $F_{U(c)}$ *and observed a value of* 1 *(one success).*

Note that the same ex ante evaluation of two different alternatives may be based on different information. For instance, the pair (u_0, C_0) *given by*

$$u_0 = (1, 0.6, 0.6) \quad \textit{and} \quad C_0 = (1, 1000, 5)$$

is interpreted as follows: the DM took 1 *sample from* $F_{U(a)}$ *and observed one success —the DM has almost no information about* $u_0(a)$*,* 1000 *samples from* $F_{U(b)}$ *and observed six-hunded successes and four-hundred failures —the DM has a lot of information about* $u_0(b)$*,* 5 *samples from* $F_{U(c)}$ *and observed three successes and two failures —the DM has some information about* $u_0(c)$*.*[4] *It seems safe to assume that the DM has no confidence in* $u_0(a)$*, lots of confidence in* $u_0(b)$*, some confidence in* $u_0(c)$*. The fact that the observation pseudocount* $C_0(x)$ *represents a degree of confidence about the ex ante evaluation* $u_0(x)$ *of* $x \in \{a, b, c\}$ *can be formalized in a Bayesian way by means of a product of beta priors, but this goes beyond the scope of the present note.* □

Given (u_0, C_0), if an alternative a is selected again and another sample from $F_{U(a)}$ is taken, a new realization $v(a)$ of $U(a)$ is observed. The empirical evaluation of a becomes

$$u_1(a) = \frac{C_0(a)}{C_0(a)+1} u_0(a) + \frac{1}{C_0(a)+1} v(a)$$

and of course $C_1(a)$ is now $C_0(a)+1$. For all $b \neq a$ in A, **empirical evaluation** and **observation count** do not change, that is $u_1(b) = u_0(b)$ and $C_1(b) = C_0(b)$.

The more observations from a are taken, the better the estimate of the expectation of $U(a)$ becomes. This **ergodic property** is at the basis of our algorithm.

Ergodic Annealing: Let $\beta_n \to \infty$.

Initialize. Set $u \doteqdot u_0$ and $C \doteqdot C_0$.

Step 0**.** Choose $a \in A$ randomly.

Update. Observe $U(a)$

$$\text{set} \quad u(a) \doteqdot \frac{C(a)}{C(a)+1} u(a) + \frac{1}{C(a)+1} U(a) \quad \text{and} \quad C(a) \doteqdot C(a) + 1.$$

Step $n+1$**.** Choose $b \in A$ randomly.

- If $u(b) \geq u(a)$, then set $a \doteqdot b$.
- If $u(b) < u(a)$, then set $a \doteqdot b$ with probability $e^{\beta_n [u(b) - u(a)]}$.

Update. Observe $U(a)$

$$\text{set} \quad u(a) \doteqdot \frac{C(a)}{C(a)+1} u(a) + \frac{1}{C(a)+1} U(a) \quad \text{and} \quad C(a) \doteqdot C(a) + 1.$$

4. Again, this "information" can be factual or introspective.

The only difference with Simulated Annealing is the update routine that allows the algorithm to learn the values as it explores.

From a purely conceptual point of view one could think of first learning the values of alternatives, then performing a standard Simulated Annealing procedure. But this would be unfeasible even in cases with relatively few alternatives (say a Traveling Salesman Problem with 40 cities). In fact, just performing one draw from each distribution would be extremely costly ($40!$ draws, one for each possible tour), and performing sufficiently many draws until the empirical averages converge is utterly impossible.

Our approach learns the payoffs and optimizes it simultaneously. This speeds up the search because the DM is not interested in finding the true payoff of all alternatives, but only an alternative with highest true payoff. In the simulations below this corresponds to the fact that while the sequence u_n of Step n evaluations does not necessarily converge, the payoff of the chosen alternative converges to the true optimal payoff. Our agent is an empirical optimizer, not an empirical statistician.

4 SIMULATIONS

In this section we benchmark the Ergodic Annealing algorithm for two classical combinatorial problems: the Directed Steiner Tree problem on graphs (DST), our initial example, and the Traveling Salesman Problem (TSP), in which the randomness of traffic and viability between two nodes makes compelling the uncertainty in the cost function.

For the DST problem we adapted the Simulated Annealing algorithm of Osborne and Gillett (1991), while for the TSP we adapted the original Simulated Annealing algorithm of Kirkpatrick et al. (1983). As discussed above, by "adapting," we mean augmenting it with an ergodic updating procedure.

4.1 Directed Steiner Tree

The first motivating example for the Ergodic Annealing algorithm is the one mentioned in the introduction and pictured in Figure 1, which is formally known as the Directed Steiner Tree problem.

In a DST problem, a directed graph $G(V, E)$ with a non-negative cost $c(e)$ associated to each edge $e \in E$ is considered. The objective is sending a packet from a root node r to each of the terminal nodes R, at *minimum cost*. Each of the $|R|$ packets is allowed to travel through some intermediate nodes, called Steiner nodes. The cost of the whole operation is the sum of the costs of the edges used to send all the packets of information, and the goal is to minimize this quantity.[5] The subset of Steiner nodes used is a variable in this problem, and the optimal configuration coincides with the minimum spanning tree of the subgraph of G induced by the root r, the terminals R, and an optimal subset of intermediate nodes.

Coherently with Figure 1, we consider networks with a *layered structure*, meaning that the vertices can be divided in an ordered partition $\{V_l\}_{l=0}^{L}$, with the singleton $\{r\}$ making up the first layer V_0 and the set of terminal nodes R making up the last layer V_L. Every edge $e = (v, w)$ in the graph must be such that its vertices are in subsequent layers, that is, $v \in V_i$ and $w \in V_{i+1}$, for some $i \in \{0, ..., L-1\}$.[6]

This is a highly non-trivial combinatorial optimization problem (indeed, it is NP-hard), and Simulated Annealing is a very successful approximation scheme used to tackle it. Therefore a DST with unknown costs is a natural candidate to test the performance of Ergodic Annealing.

5. Note that we can use an edge as an intermediate channel to reach two different terminal nodes, but its cost will be counted only once. This can be interpreted as having only a fixed "opening" cost of the channel and no capacity constraints.

6. Since the graph is directed, the order of the vertices is important.

A key step required to run an annealing algorithm for this problem is the selection of feasible moves from one candidate solution to another.[7] There are different ways to do this. We follow the proposal of Osborne and Gillett (1991), which simply consists in allowing to move one potential Steiner node ($v \in \bigcup_{l=1}^{L-1} V_l$) from the set of used Steiner nodes to the set of unused nodes, and *vice versa*. Then one can easily compute the new Steiner tree by computing the minimum spanning tree on the resulting subgraph by using Edmonds' algorithm.[8]

To study and compare the performance of Ergodic Annealing with respect to Simulated Annealing we ran both algorithms on a test set of 1000 random graphs with the same true costs each time —**known** for the Simulated Annealing agent, **unknown** to the Ergodic Annealing one. Each graph $G = (V, E)$ in the test set has 13 layers (so 11 layers of potential Steiner nodes), with a maximum of 12 nodes for each non-root layer. The actual number of nodes for each layer is chosen uniformly at random from $\{2, ..., 12\}$. For each $v \in V$, a node from the previous layer is selected randomly and automatically connected, to guarantee feasibility. All other possible edges in the graph are present with probability $1/2$. The (true) arc costs are drawn uniformly from the interval $(0, 1)$, and the ex ante expected cost of each arc is initialized to $1/2$.

Since root and terminals are fixed in the optimal Steiner tree, we define the *size* of the graph as the number of potential Steiner nodes. The average size of a graph in the test was 71.3.

In these moderately large graphs, the two algorithms performed quite similarly. Indeed, they reached the same final configuration on 322 graphs, and the average absolute deviation with respect to the best configuration found[9] was 0.04664. In words, on average the two procedures found solutions whose costs differed by 4.66%, sometimes with Simulated Annealing being closer to the true optimal solution, sometimes with Ergodic Annealing performing better.

In Figure 2 we present two examples of Steiner trees found by Ergodic Annealing and Simulated Annealing on graphs from the test set.

Ergodic Annealing finds configurations of similar cost compared Simulated Annealing, even if the problem it faces is orders of magnitude harder. Simulated Annealing optimizes over a large but finite set of **known** configuration costs, while Ergodic Annealing searches for a minimum cost configuration on a space that is potentially infinite, because the true costs are **unknown** and are *learned* on a continuum space.

4.2 Traveling Salesman Problem

The second benchmark studied in this paper is the well-known Traveling Salesman Problem.

In the classical case a list of cities and distances between each pair of cities are given and known, and the objective is to find the shortest possible route that visits each city and returns to the starting point. Just like the DST, the TSP is an NP-hard problem in combinatorial optimization, important in theoretical computer science, operations research and economics.

In our variant, distances are replaced by average travel times, and the objective is finding the fastest route. The Simulated Annealing algorithm can solve this problem when these travel times are **known**, the Ergodic Annealing can solve it even when travel times are **unknown**.

We ran a simulation over 2000 random instances of TSP, with cities location chosen randomly from the unit square. The number of cities was selected at random between 30 and 90, with an average size of graphs in the simulation of 59.68. The

7. A move is considered feasible if it transforms a feasible solution into another feasible solution.

8. In this layered version computing the minimum arborescence is particularly efficient, because the graph is a DAG and there are no recursive calls in the algorithm.

9. The best configuration is the one of lower cost among the two final configurations found by the algorithms.

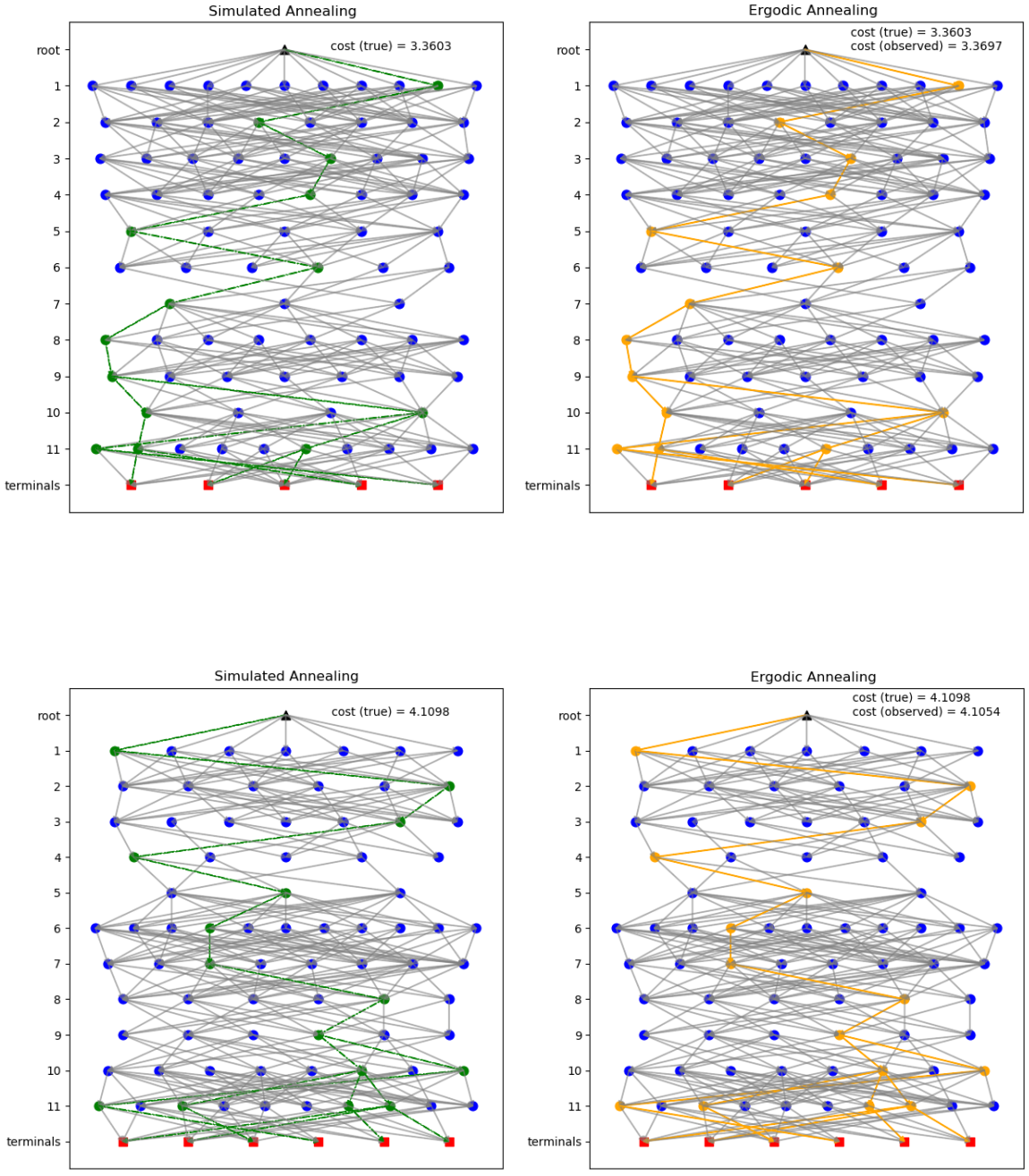


Fig. 2. Comparison of Ergodic Annealing and Simulated Annealing on two random DST instances.

performance of the two algorithms on the test was almost identical, with Ergodic Annealing performing at least as well as Simulated Annealing on 995 graphs. The average absolute deviation with respect to the best configuration found was 1.90%.

This simulation provides an even stronger evidence than the one found with the previous benchmark about the validity of Ergodic Annealing.

In Figures 3 and 4 we present two examples of optimal routes found by Ergodic Annealing and Simulated Annealing

on graphs from the test set.

5 Conclusion

For a given expected payoff u, Ergodic Annealing (implementable by a DM who ignores u and must learn it from the environment) performs almost as well as Simulated Annealing (which requires the DM to know u ex ante). Thus, Ergodic Annealing seems to be a promising extension of Simulated Annealing to decision making under uncertainty.

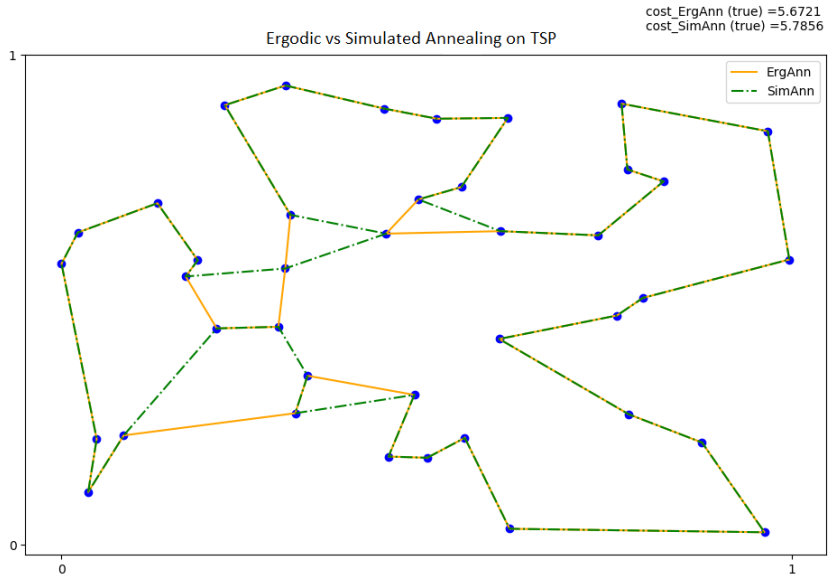


Fig. 3. Random TSP instance with 40 cities where Simulated Annealing finds a slightly suboptimal route compared with Ergodic Annealing.

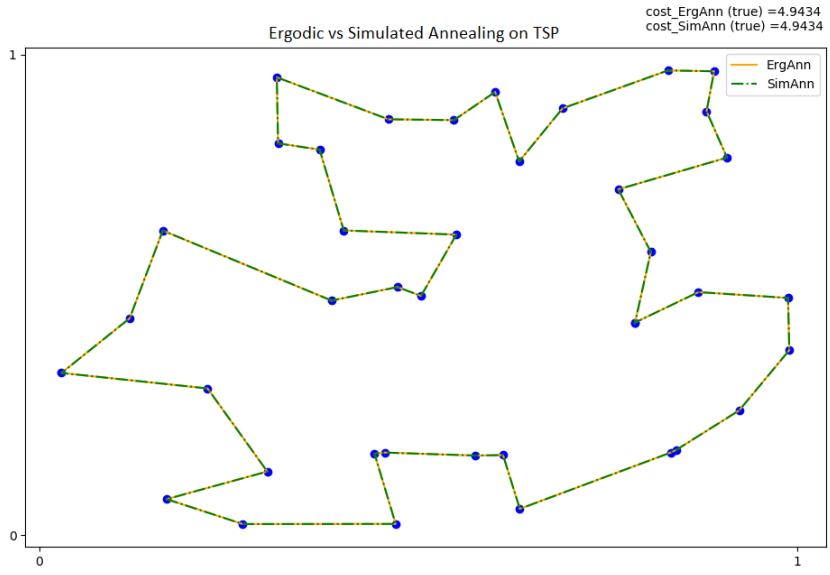


Fig. 4. Random TSP instance with 40 cities that produced the same final configuration with both algorithms

REFERENCES

[1] Edmonds, J. (1967). Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71.4, 233-240.

[2] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97-109.

[3] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.

[4] Madras, N. N. (2002). *Lectures on Monte Carlo methods*. American Mathematical Society.

[5] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087-1092.

[6] Osborne, L. J., & Gillett, B. E. (1991). A comparison of two simulated annealing algorithms applied to the directed Steiner problem on networks. *ORSA Journal on Computing*, 3, 213-225.

Noise Impact On Convolutional Neural Networks Ability to Generalize

Abdeldjallil AIDI[1], Karim MEZZOUG[2] and Abou EL Hassane BENYAMINA[3]

1. Ahmed Ben Bella University of Oran, Algeria
aidi.abdeldjallil@edu.univ-oran1.dz
2. Ibn Khaldoun University of Tiaret, Algeria
karim.mezzoug@univ-tiaret.dz
3. Ahmed Ben Bella University of Oran, Algeria
benyamina.hassen@univ-oran1.dz

Abstract. In this paper, we try to clarify the effects of noise contained in the images within image classification tasks by analyzing two different types of noise (Salt and Pepper, Gaussian) with five different levels on three Convolutional Neural Network (CNN) models (XceptionNet, GoogleNet, ResNet) using the same parameters (Dataset, noise, and level of noise), and how denoising methods can help to alleviate this problem, for the last matter, we consider two algorithms the median filter for the salt and pepper noise, and the non-local means (NLM) for the gaussian noise.
We perform our experiments with the Cifar-10 dataset, our results show that noise in images can hinder classification tasks and cause it a problem (make it harder to separate classes). Although images were denoised, we were unable to reach the results obtained in the noise-free scenarios.

Keywords: Classification, Noise, Denoising, Convolution, Neural Network

1 Introduction

According to literature, nearly 90% of the information received by humans is visual. Hence the production of quality images, as well as their digital (and if possible) automatic processing, is therefore of considerable importance, while most of computer vision and classification systems neglect pre-processing [08] and assume that images are given with high quality [04].

In the internet, in our phones and laptops, Millions of pictures ranging from biomedical images to the images of natural surroundings, such pictures might contain a lot of important information in diverse domains of application, which represent a primary source of information and/or visualization. The quality of the output image may be inferior to that of the original input picture when converted from one form to another by processes like imaging, scanning, or transmitting. Hence, there is a need to improve the quality of such images, so that the output image is better for human perception or machine analysis.

Nowadays Image classification is used in various applications, such as agricultural [01], educational and medical [05]. Our work is a part of the classification of images using deep learning which is a learning technique that enables a program, for example, to understand spoken language or recognize the content of an image allowing machines to learn and recognize objects, now back to our work, it is about of a set of comparisons between three architectures of convolutional neural networks. We aim to create a certain number of classifiers to images that contains noise and their restored versions, the results will be compared with each other to see the resilience of Convolutional Neural Networks (CNN) over distinct type of noise with different levels.

2 Related works

This is not the first time such experimentation is done to understand the impact of noisy images (in any way possible) in the performance of CNNs within image classification tasks. There are papers studying the effects of noise in the capability of CNNs to learn [02], [07].

Noise in images can hinder classification tasks; this knowledge is already discovered with systems that employ convolutional Neural Networks [06], and in systems that use handcrafted features. In [04] They evaluated 4 deep neural network architectures, they show that the performance of these networks is affected when classifying images with lower quality compared with the image quality in the training set, their experiments do not cover the presence of noisy images in the training set.

While Paranhos da Costa et al in [03] takes in consideration that noisy images might appear in the training set, they created noisy versions and generated their restored versions, they used hand-crafted features (LBP and HOG) and SVM classifiers were trained with each version of the training set. Their results show that classifiers suffer to generalize to different noisy data and image classification becomes harder.

Our work is an extension of [03]. As all we believe that noise in any way possible, makes classification more difficult, our experiments exploited state of the art deeper artificial neural network architectures such as RESNET and XCEPTIONET, second, we train all the architectures using the same dataset version for better comparison.

3 Results and Discussion

To evidently visualize the impact of noise and filtering methods in the image quality, the structural similarity index measure (SSIM) [09] and the peak signal-to-noise ratio (PSNR) [10] values are shown in **TABLE I**.

Smaller values in both the PSNR and SSIM indicate less similarity between the images of the original dataset version and the version compared to. By comparing the results while training and testing our chosen models with the same dataset (same noise and level) we can see how much harder separating between classes gets for these models. It has long been clear that increasing noise levels have undesired consequences. To better understand this, **TABLE II** shows the accuracy of our selected models trained on noisy datasets and their restored versions.

			SSIM		PSNR	
			noisy	filtered	noisy	filtered
Cifar-10 Data set	Salt and Pepper	P = 0.1	0.76	0.97	15.26	25.39
		P = 0.2	0.60	0.95	12.46	23.47
		P = 0.3	0.48	0.93	10.90	21.19
		P = 0.4	0.41	0.88	9.85	18.92
		P = 0.5	0.34	0.82	9.08	16.87
	Gaussian	σ = 10	0.87	0.87	16.92	17.17
		σ = 20	0.73	0.77	14.22	15.50
		σ = 30	0.58	0.67	12.45	15.29
		σ = 40	0.44	0.54	11.20	14.62
		σ = 50	0.31	0.45	10.28	13.77

TABLE I SSIM AND PSNR FOR EACH NOISE LEVEL

			XceptionNet		GoogleNet		ResNet	
			noisy	filtered	noisy	filtered	noisy	filtered
Cifar-10 Data set	original		85.89		84.96		77.46	
	Salt and Pepper	P = 0.1	77.12	81.15	77.98	80.12	66.07	73.04
		P = 0.2	74.47	78.50	72.11	77.84	62.89	72.38
		P = 0.3	66.26	76.14	67.59	75.61	58.22	69.38
		P = 0.4	62.77	72.88	62.48	74.80	51.43	67.33
		P = 0.5	57.02	74.43	57.06	73.30	49.04	64.94
	Gaussian	σ = 10	76.76	69.92	73.70	68.02	68.11	63.29
		σ = 20	75.24	70.45	73.82	71.42	68.21	63.61
		σ = 30	60.37	62.54	66.58	58.97	62.31	56.98
		σ = 40	52.33	55.69	51.44	44.99	47.32	52.15
		σ = 50	27.97	51.54	25.60	51.28	23.19	46.83

TABLE II PERCENTAGE ACCURACY OF EACH MODEL WHEN TRAINING AND TESTING USING THE SAME DATASET VERSION

our scrutiny is divided into two steps (in both steps we chose the middle level of noise to work with throughout this scrutiny):

A. Inter-models: The goal here, is to visualize which architecture is better resilient when dealing with noise in datasets, and see how much harder classifying these image datasets gets. Even so it seems that GoogleNet did the great work but in general, and accordingly to **TABLE II** the XceptionNet model did better than the other models when dealing with noisy dataset.

B. Intra-models: Here we compare the obtained accuracies using the same model with three datasets (noise-free, noisy, denoised) the goal is to see how the image quality affect the models, and how much the denoising methods are helping in the cases that of noisy images. To display the importance/effects of de-

noising methods, and how they are helping override this hurdle the results shown in **TABLE II** are plotted in **Fig**. 1 and **Fig**. 2.

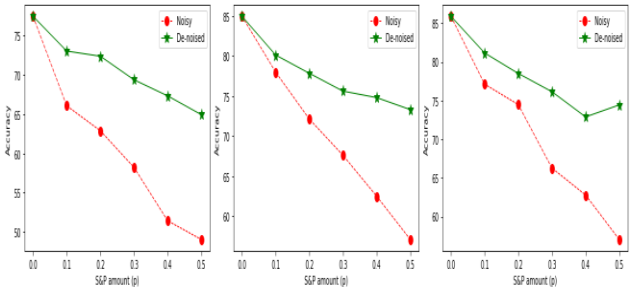


Fig. 1. Comparison of the accuracy of each network while training with Salt and Pepper noisy datasets and de-noised ones using median filter (a) results using ResNet architecture (b) results using GoogleNet architecture (c) results using XceptionNet architecture.

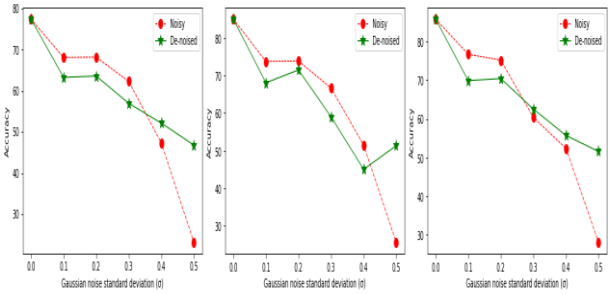


Fig. 2. Comparison of the accuracy of each network while training with Gaussian noisy datasets and de-noised ones using NLM filter (a) results using ResNet architecture (b) results using GoogleNet architecture (c) results using XceptionNet architecture.

4 Conclusion

Training deep convolutional neural networks with datasets after adding noise to their images at a known level has shown how hard classifying these images gets, our study covered two types of noise with five levels for each type, likewise you can test other types of noise with different models. Our results show that the CNN architectures are having a hard time classifying the images affected by noise in both training and test sets.

Regarding the noise reduction algorithms, the dataset restored by median filtering where able to improve accuracy and image quality compared to their Salt and Pepper noisy counterparts, but in the other hand, datasets versions that were hindered with Gaussian noise gave better results than their restored counterparts using NLM filter, that's because of the resulted blur in the image after using the NLM filter.

The final point of this paper is that the more noise you add to an image, the harder it is to classify.

References

[01] Gill, H. S., Khalaf, O. I., Alotaibi, Y. Alghamdi, S. & Alassery, F. (2022). Fruit Image Classification Using Deep Learning. Computers, Materials and Continua, 71(2), 5135–5150.

[02] AJ. Bekker and J.Goldberger(2016). Training deep neural networks based on unreliable labels.

[03] GB. P. da Costa, WA. Contato, TS. Nazare, JE. S. BNeto, and M. Ponti (2016). An empirical study on the effects of different types of noise in image classification tasks.

[04] S. Dodge and L. Karam (2016). Understanding how image quality affects deep neural networks.

[05] AH. Khan, S. Abbas,MA. Khan, U. Farooq, WA. Khan, SY. Siddiqui, and A. Ahmad (2022). Intelligent Model for Brain Tumor Identification Using Deep Learning. Applied Computational Intelligence and Soft Computing.

[06] M. Momeny, Ali M. Latif, M. Agha Sarram, R. Sheikhpour, YD. Zhang (2021), A noise robust convolutional neural network for image classification, Results in Engineering, Volume 10.

[07] DF. Nettleton, A. Orriols-Puig, and A. Fornells (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial Intelligence Review, 33.

[08] M. Ponti, TS. Nazaré, and GS. Thumé (2016). Image quantization as a dimensionality reduction procedure in color and texture feature extraction. Neurocomputing, 173.

[09] De Rosal Igantius Moses Setiadi (2021). Psnr vs ssim: imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80.

[10] Z. Wang, AC. Bovik, HR. Sheikh, and Eero P. Simoncelli (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13.

Implementation of Metaheuristic Search for Finding Shortest Path in Tunnelled Network

Habiba Akter*[1]

[1]Queen Mary, University of London – United Kingdom

Abstract

This work investigates the implementation of an Evolutionary Algorithm (EA) as a metaheuristic search tool to design a path computation tool to help find the shortest path(s) from a source node to a destination node to send data over. The bespoke tool will also provide an additional benefit of loose source routing in a scenario where internet tunnels are present in a part of the network topology. However, the EA-generated optimal paths may or may not have tunnels present in them but the costs associated will be optimised using the evolutionary computation. The objective functions for optimisation will be designed based on the cost values associated with the end-to-end path.

*Speaker

Feature Selection Problem: a short slr about its formulation

José Barrera*[1], Broderick Crawford*[1], Felipe Cisternas[1], Ricardo Soto[1], and Giovanni Giachetti[2]

[1]Pontificia Universidad Católica de Valparaíso – Chile
[2]Universidad Andrés Bello [Santiago] – Chile

Abstract

This extended abstract presents the methodology employed in a systematic literature review (SLR) that aims to investigate Feature Selection (FS) and Optimization Techniques. The study focuses on understanding the objective function, performance evaluation metrics, optimization techniques, and practical applications of feature selection. The methodology section provides an overview of the systematic approach adopted for conducting the literature review.

*Speaker

Impact of Structural Bias on the Sine Cosine Algorithm: A Theoretical Investigation Using the Signature Test

Kanchan Rajwar*[1], Kusum Deep[1], and Muthu Mathirajan[2]

[1]Department of Mathematics, Indian Institute of Technology Roorkee, Uttarakhand – India
[2]Department of Management Studies, Indian Institute of Science, Bangalore – India

Abstract

Metaheuristic algorithms have been recognized for their effectiveness in solving non-convex and non-linear complex optimization problems. These algorithms are influenced by landscape bias, guided by objective function values, and algorithmic operator bias directed by the operator used in the algorithms. The presence of algorithmic operator bias, also known as structural bias, forces the population to revisit a particular region, badly affecting the algorithm's exploration capacity. Also, since the population revisits the same place without gaining new information, it increases computational costs and slows the convergence rate. Therefore, it is crucial to identify and address structural bias to enhance algorithm performance and reduce computational time. To the best of our knowledge, no previous study has focused on investigating the structural bias of the Sine Cosine Algorithm (SCA) in the existing literature. Therefore, the main objective of this study is to examine the structural bias present in the SCA, a widely used metaheuristic algorithm. To investigate structural bias signature test is employed. Additionally, average Euclidean distances of the population is calculated to assess spatial relationships and overall distribution. Our analysis uncovers a prominent bias in the SCA towards the axes and the origin, suggesting a strong tendency to converge towards specific regions within the search space. By understanding and characterizing this bias, we provide valuable insights into the behavior of the SCA, which can contribute to the research community's understanding and guide future improvements in algorithm design.

*Speaker

A Study About Meta-Optimizing the NSGA-II Multi-Objective Evolutionary Algorithm

José F. Aldana-Martín*[1], Antonio J. Nebro[1,2], Juan J. Durillo[3], and María Del Mar Roldán García*[1,2]

[1]ITIS Software, University of Málaga, 29071, Málaga – Spain
[2]Departamento de Lenguajes y Ciencias de la Computación. University of Málaga, 29071 Málaga – Spain
[3]Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities – Germany

Abstract

The automatic design of multi-objective metaheuristics is an active research line aimed at, given a set of problems used as training set, to find the configuration of a multi-objective optimizer able of solving them efficiently. The expected outcome is that the auto-configured algorithm can be used of find accurate Pareto front approximations for other problems. In this paper, we conduct a study on the meta-optimization of the well-known NSGA-II algorithm, i.e., we intend to use NSGA-II as an automatic configuration tool to find configurations of NSGA-II. This search can be formulated as a multi-objective problem where the decision variables are the NSGA-II components and parameters and the the objectives are quality indicators that have to be minimized. To develop this study, we rely on the jMetal framework. The analysis we propose is aimed at answering the following research questions: RQ1 - how complex is to build the meta-optimization package?, and RQ2 - can accurate configurations be found? We conduct an experimentation to give an answer to these questions.

*Speaker

An optimised version of differential evolution heuristic for feature selection

Thibault Anani*[1], Francois Delbot[1,2], and Jean-François Pradat-Peyre[1,2]

[1]LIP6 – Sorbonne Universite, Centre National de la Recherche Scientifique, Centre National de la Recherche Scientifique : UMR7606 – France

[2]Université Paris Nanterre - Département de Mathématiques et Informatique – Université Paris Nanterre – France

Abstract

Feature selection plays a crucial role in optimizing the quality of machine learning models by identifying relevant features while discarding redundant and harmful ones. However, the combinatorial explosion of feature subsets makes exhaustive evaluation impractical. In this article, we focus on feature selection for Amyotrophic Lateral Sclerosis (ALS) prediction using clinical trial data. We also evaluate different selection methods in challenging scenarios, highlighting the impact of dataset structure. With limited, noisy, and redundant data, we propose an improved heuristic based on differential evolution called Tournament In Differential Evolution (TIDE). Experimental results demonstrate that our heuristic outperforms common feature selection techniques, especially in ALS data achieving a predictive accuracy score of 82.04%. Our hybrid heuristic, "TIDE+ReliefF," consistently achieves the best predictive quality, convergence speed, and stability while converging on a smaller feature subset. Overall, our study emphasizes the contribution of feature selection methods, particularly differential evolution, in improving predictive quality not only for ALS but for most datasets regardless of their structure.

*Speaker

A Honey Bee Mating Optimization HyperHeuristic for Patient Admission Scheduling Problem

Imen Oueslati*[1,2], Moez Hammami[1], Issam Nouaouri[2], Ameni Azzouz[1], Lamjed Bensaid[1], and Hamid Allaoui[2]

[1]Université de Tunis, Institut Supérieur de Gestion de Tunis, Smart-Lab – Tunisia
[2]LGI2A – Université d'Artois : Faculté des Sciences Appliquées – France

Abstract

Hyperheuristics represent a generic method that provides a high-level of abstraction so as to be able to deal with several problem domains. This category of methods consists in managing a set of heuristics and tries to find the best sequence that gives good quality results. This paper proposes a hyperheuristic simulating the honey bees mating behavior to solve the Patient Admission Scheduling Problem (PASP). The PASP is an NP-hard problem that represents an important field in the health care discipline. To perceive the influence of the heuristics on solving the problem, we implemented two versions of hyperheuristics, each one is working on a different set of heuristics. The results show that one of the versions generates better results than the other which reveals the important role of the quality of basic heuristics to enhance the hyperheuristic performance.

*Speaker

Metaheuristic Algorithms for Circle Packing Problem: A Comprehensive Review

Yogesh Kumar*[1] and Kusum Deep*[1]

[1]Indian Institute of Technology Roorkee, Roorkee – India

Abstract

The Circle Packing Problem (CPP) is a well-known optimization problem with a wide range of applications. It is difficult to identify exact solutions for the CPP as it is a NP-hard problem. As a result, metaheuristic algorithms are efficient approach to address this challenging problem. This paper presents an extensive literature review of the role and effectiveness of metaheuristic algorithms in solving the CPP. The objective of this paper is to examine the applications, advancements, and potential of metaheuristic algorithms in solving the CPP. Moreover, case studies and real-world examples of how metaheuristics have been used to address the CPP. Finally, some suggested possible future areas are also highlighted for the researchers who want to utilize metaheuristic algorithms for solving the CPP and related optimization problems.

*Speaker

Opposition Based Local Escaping Marine Predators Algorithm for Continuous Optimization

Manish Kumar*[1] and Kusum Deep*[1]

[1]Indian Institute of Technology Roorkee – India

Abstract

The marine predators algorithm (MPA) is a recently developed metaheuristic algorithm that is inspired by the foraging behavior of marine predators. It has been widely used to solve real-life optimization problems. However, it frequently gets trapped in a local optima since it is unable to have a diversified population in the early stages of optimization. To overcome the premature convergence problem of MPA, this paper introduces an improved version of the marine predators algorithm named as opposition-based local escaping marine predators algorithm (OLMPA). There are two ways in which the improvement is carried out. The first improvement uses opposition based learning (OBL) assures the diversity of solutions in the search space. The second improvement uses the local escaping operation, which creates new solutions that replace the worst solutions to estimate the best solution. These enhancements are designed to address the imbalance between exploration and exploitation. The proposed OLMPA is tested on 23 benchmark functions. The numerical and statistical experimental results show that the proposed algorithm overperformed classical MPA.

*Speaker

Harnessing Collective Intelligence: Integrating Particle Swarm Optimization and Reinforcement Learning for Efficiently Solving Binarized Coverage Problems

Marcelo Orlando Becerra Rozas*[1], Broderick Crawford[1], Ricardo Soto[1], El-Ghazali Talbi[2], and Giovanni Giachetti[3]

[1]Pontificia Universidad Católica de Valparaíso – Chile
[2]University of Lille INRIA – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France
[3]Universidad Andres Bello – Chile

Abstract

This article proposes the utilization of a classical metaheuristic in our framework for selecting binarization schemes based on reinforcement learning. Our selector enables the binarization of continuous metaheuristic, allowing for its application in binary domains. In our previous works, reviewers commonly questioned why we didn't use more popular metaheuristics like PSO. Finally, we present the implementation and the results obtained, which demonstrate the successful performance of PSO, whether using QL or BQSA, in solving a coverage problem.

*Speaker

Network flow models for days off scheduling

Ghassen Tlig*[1] and Fethi Jarray[2]

[1]ESEO - – École supérieure d'électronique de l'ouest [Angers] – France
[2]LIMTIC Laboratory, UTM University – Tunisia

Abstract

This paper studies the days off scheduling problem when the demand for staffing may differ from day to another and when the total load is fixed in advance for each employee. The scheduling problem is then to assign on-days and days-off to employees with different objectives: (1) exactly met the demand and the offer requirement (2) satisfy as best as possible the requirements. For each one, we propose a polynomial time algorithm based on network flow to construct a feasible scheduling.

*Speaker

High Performance Algorithms for the Unrelated Parallel Machines Scheduling Problem with a Common Server and Job-Sequence Dependent Setup Times

Youssouf Hadhbi*[1], Laurent Deroussi , Nathalie Grangeon , Sylvie Norre , and Christophe Blanchon

[1]Laboratoire dÍnformatique, de Modélisation et dÓptimisation des Systèmes LIMOS – Université Clermont Auvergne, CNRS – 1 Rue de la Chebarde, Aubiere, 63178 Clermont Ferrand, France

Abstract

This paper deals with a new variant of the non-preemptive unrelated parallel machines scheduling problem with setup times. We first formulate the problem as a mixed integer linear program (MILP), and further devise an exact algorithm based on a branch-and-cut (B&C) algorithm for solving the problem. Due to the complexity of the problem, we propose a metaheuristic based on an iterated local search (ILS) algorithm. Using this, we provide several matheuristics based on a two stage algorithm: ILS first and branch-and-cut last. We carry out a comparative study between these methods and show the effectiveness of our approach using small-sized instances and large-sized ones.

*Speaker

Uniformly Deployed Sets in Computer Science and Optimisation

Peter Czimmermann*[1] and Zuzana Borcinova[2]

[1]University of Zilina – Univerzitna 1, 010 26 Zilina, Slovakia, Slovakia
[2]University of Zilina – Slovakia

Abstract

Uniformly deployed sets (UDS) are combinatorial objects that contain n-bit binary words of weight p as their elements and each pair of such words has at most t ones on the same positions (where t is a given natural number). In our contribution, we present a fast algorithm for the construction of UDS and we show some possibilities of their usage in various areas of computer science. For example, UDS can be used for increasing the effectiveness of heuristics in optimisation problems.

*Speaker

Cluster images with AntClust: a clustering algorithm based on the chemical recognition system of ants

Winfried Oed*[1] and Parisa Memarmoshrefi[1]

[1]Georg-August-University = Georg-August-Universität Göttingen – Germany

Abstract

We implement AntClust, a clustering algorithm based on the chemical recognition system of ants and use it to cluster images of cars.
We will give a short recap summary of the main working principles of the algorithm.
Further, we will describe how to define a similarity function for images and how the implementation is used to cluster images of cars from the vehicle re-identification data set.
We then test the clustering performance of AntClust against DBSCAN, HDBSCAN and OPTICS.
Finally one of the core parts in AntClust, the rule set can be easily redefined with our implementation, enabling a way for other bio-inspired algorithms to find rules in an automated process.
The implementation can be found on GitLab "https://gitlab.com/Winnus/antclust"

*Speaker

EVADyR: a new dynamic resampling algorithm for optimizing noisy expensive systems

Soraya Zertal[1], Sophie Robert*[2], and Philippe Couvee[2]

[1]Li-PaRAD lab. Univ UPSaclay – UPSaclay – France
[2]ATOS BULL – ATOS BULL, Atos-Bull – France

Abstract

Black-box auto-tuning methods have been proven to be efficient for tuning configurable computer appliance. However, because of the shared nature and the complexity of the software and hardware stack of some systems such as cloud or HPC systems, the measurement of the performance function can be tainted by noise during the tuning process, which can reduce and sometime prevent, the benefit of auto-tuning. An usual choice is to add a resampling step at each iteration to reduce uncertainty, but this approach can be time-consuming. In this paper, we propose a new resampling and filtering algorithm called EVADyR (Efficient Value Aware Dynamic Resampling). This algorithm is able to tune efficiently a prefetching strategy in the case of multiple parallel accesses. Because it finds a better exploration versus exploitation trade-off by resampling only promising parametrizations and increases the level of confidence around the suggested solution as the tuning process advances, it outperforms state of the art dynamic resampling by reducing the distance to the optimum by 93.5%, as well as speed-up the experiment duration by 45.8% because less iterations are needed to reach the found optimum. An additional proof of this study is the demonstration of the importance of using noise reduction strategies for the optimization of highly shared resources such as HPC or cloud systems.

*Speaker

Multi-Objective Metaheuristic Solution Approach for the Crop Plant Scheduling Problem

Maksim Lalić*[1], Obrenović Nikola[1], Selin Ataç[2], Stefano Bortolomiol[3], Marko Oskar[1], Sanja Brdar[1], Vladimir Crnojević[1], and Ivan Luković[4]

[1]Institute BioSense, University of Novi Sad – Serbia
[2]Ecole Polytechnique Fédérale de Lausanne, Transport and Mobility Laboratory (TRANSP-OR) – Switzerland
[3]Optit Srl, Bologna – Italy
[4]Faculty of Organizational Sciences, University of Belgrade – Serbia

Abstract

Accurate and optimized harvest schedules should improve the subsequent harvest performance. In this work, we formulate a bi-objective optimization model used to improve harvest effectiveness, measured by the amount of unnecessarily produced waste, and its efficiency, measured by the time required for harvest completion. The proposed model considers a farm field divided into management units that are treated and harvested independently and assume storage and processing limits that dictate the pace of the optimal harvest. The optimization problem is solved using 2 distinctive metaheuristic techniques. We examined adaptive large neighborhood search (ALNS) as a model-driven single-solution-based metaheuristic and non-dominated sorting genetic algorithm II (NSGA-II) as a data-driven population-based approach. Optimization techniques were tested on 20 synthesized test cases. All obtained results emphasize the significant and consistent dominance of the ALNS over the NSGA-II.

*Speaker

Memetic Algorithms for the Workforce Routing and Scheduling Problem: case study of Energy Distribution System Operator (DSO)

Loïc Cardinaël*[1,2], Wahiba Ramdane Cherif-Khettaf[1], and Ammar Oulamara[1]

[1]Laboratoire Lorrain de Recherche en Informatique et ses Applications – Université de Lorraine – France

[2]efluid sas – efluid sas – France

Abstract

This study addresses a real industrial problem faced by a company operating an energy distribution network: the efficient assignment of tasks to their agents.
This problem involves determining the best routes for agents to visit multiple tasks, known in the literature as the Technician Routing and Scheduling Problem (TRSP).

It considers a limited number of agents, and tasks located in different areas. Each task has a service duration, a time window during which the service should be executed, and a set of required skills.

Additionally, agents possess different skills, multiple availability intervals, and a maximum daily working time. The objective is to minimize the total traveled distance, including penalties for unrouted tasks.
To tackle this problem, we propose an order-first split-second approach that combines a memetic algorithm using a giant tour encoding of the chromosome with an extension of the optimal split method. Our primary objective is to evaluate the effectiveness of this approach on a real case of the TRSP problem in the context of energy network management. The second objective is to compare our genetic algorithm, which uses the giant tour encoding with a genetic algorithm scheme that employs an indirect encoding representation proposed in the literature. Il is worth noting that this indirect enconding has proved its effectiveness on real instances of an industrial problem, which is similar to our case study. We aim to test different implementations of the proposed genetic algorithms on real instances to evaluate the impact of the extended optimal split, local search, and encoding type.

*Speaker

A comparative analysis of simheuristics to address the berth allocation problem in bulk ports

Daniel Mendoza-Casseres*[1]

[1]Universidad del Atlántico – Colombia

Abstract

This study deals with the Berth Allocation Problem (BAP) in bulk ports which are NP-hard problems. Bulk cargo is one that is transported in large quantities without packaging. The bulk terminals carry out the loading through devices called shiploaders. Metaheuristics such as Genetic Algorithms, Tabu Search, Greedy Randomized Adaptive Search Procedure (GRASP) and Simulated Annealing (SA) have been implemented for BAP in bulk ports. Other studies have used simulation for the BAP; however, there is little research that uses simulation-optimization approaches, mainly using the Simheuristic method. Simheuristics extend metaheuristics by adding a simulation layer that allows the optimization component to deal with scenarios under uncertainty.This short paper proposes the scheduling of vessels with the variation of shiploaders in bulk ports using GRAPS or SA metaheuristics in a simulated environment to make a comparison. Initially, the SA and GRASP metaheuristics have been implemented for the BAP in bulk ports without simulation. This extended abstract of work-in-progress proposes a simheuristic approach for addressing the berth allocation problem in bulk ports. Stochastic loading times and stochastic set up times will be considered in a bulk terminal to show the use of the resources used by the port. The simulation model discrete BAP with metaheuristics will be built using FlexSim Software 2023. An optimization cycle will be scheduled. This cycle consists of configuring an optimization loop using C++ and FlexSim. The purpose is to obtain results from stochastic instances with the Simheuristics to minimize the total penalty cost. **Keywords**: Simheuristics, Scheduling, GRASP, SA, BAP.

*Speaker

A Simulated Annealing Based Approach for the Roman Domination Problem

Jakob Greilhuber*[1], Sophia Schober[1], Enrico Iurlano[1], and Günther R. Raidl[1]

[1]Algorithms and Complexity Group, TU Wien, Favoritenstr. 9/1921, 1040 Vienna – Austria

Abstract

The Roman Domination Problem is an NP-hard combinatorial optimization problem on an undirected simple graph. It represents scenarios where a resource shall be economically distributed over its vertices while guaranteeing that each vertex has either a resource itself or at least one neighbor with a sharable surplus resource. We propose several (meta-)heuristic approaches for solving this problem. First, a greedy construction heuristic for quickly generating feasible solutions is introduced. A special feature of this heuristic is an optional advanced tiebreaker. This construction heuristic is then randomized and combined with a local search procedure to obtain a greedy randomized adaptive search procedure (GRASP). As an alternative, we further propose a simulated annealing (SA) algorithm to improve the solutions returned by the construction heuristic. As we observe different pros and cons for the GRASP and the SA, we finally combine them into a simulated annealing hybrid, which interleaves phases of greedy randomized construction and phases of simulated annealing. All algorithms are empirically evaluated on a large set of benchmark instances from the literature. We compare to an exact mixed integer linear programming model that is solved by Gurobi as well as to a variable neighborhood search from the literature. In particular the simulated annealing hybrid turns out to yield on average the best results, making it a new state-of-the-art method for the Roman domination problem.

*Speaker

Mixed-variable surrogate-based optimization using probability features for categorical variables

Charlotte Beauthier*[1], Sylvério Pool Marquez[2], Caroline Sainvitu[3], and Annick Sartenaer[2]

[1]Cenaero ASBL – Bâtiment Eole Rue des Frères Wright 29 B-6041 Gosselies, Belgium
[2]University of Namur, naxys Institute – Belgium
[3]Cenaero ASBL – Belgium

Abstract

A globally effective approach to high-fidelity optimization problems based on computationally expensive analysis lies in the exploitation of surrogate models, also known as metamodels or response surface models. They act as cheap-to-evaluate alternatives to the original high-fidelity models reducing the computational cost, while still providing improved designs. The underlying principle of Surrogate-Based Optimization (SBO) consists in accelerating
the optimization process by essentially exploiting surrogates for the objective and constraint evaluations, with a minimal number of function calls to the high-fidelity models for keeping the computational time within affordable limits.

In the literature, the vast majority of described SBO is limited to a purely continuous design space, i.e., an optimization

context where each design variable is a continuous variable.

However for real engineering problems, design variables can
have different natures (continuous, integer, discrete and categorical).
One of the main challenges is the handling of the non-continuous variables within the whole optimization process. We present here a SBO framework handling mixed variables with a focus on the
management of the categorical ones. More precisely, we propose an adaptation of the strategy that
(Wang et al., 2021) have developed to handle categorical variables within a Particle Swarm Optimization (PSO) algorithm, to an Evolutionary Algorithm (EA) as inner optimizer of the SBO loop.

Complete abstract in attachment, pdf file. thanks

*Speaker

Machine Learning-based Per-Instance Algorithm Selection for High-Performance Subgraph Isomorphism Enumeration

Syed Ibtisam Tauhidi*[1], Arindam Karmakar[2], Thai Son Mai[3], and Hans Vandierendonck[3]

[1]Queen's University Belfast – United Kingdom
[2]Tezpur University – India
[3]Queen's University Belfast – United Kingdom

Abstract

The Subgraph Isomorphism Enumeration (SIE) problem requires discovering all the embeddings of a *pattern* (or *query*) subgraph in a given *data* graph. The problem is NP-complete, and numerous exact heuristic algorithms have been proposed to speed up the execution of the problem. Based on the observation that no heuristic is the fastest for all pattern and data graph pairs, we design a metaheuristic for *per-instance algorithm selection* to determine the fastest heuristic for each graph pair. We hypothesise that the connections and properties of vertices in the graph pair are indicative of the best-performing heuristic algorithm. As such, we design a Machine Learning (ML)-based metaheuristic algorithm and investigate how well various types of graph features and ML algorithms predict performance. Our best-performing metaheuristic improves the execution speed of the SIE problem by up to 1.54 times across 8 data graphs compared to any single heuristic algorithm. The analysis furthermore identifies remaining challenges unique to specific data graphs in the SIE problem.

*Speaker

A systematic review and ranking of operators in adaptive large neighborhood search for vehicle routing problems

Stefan Voigt*[1]

[1]Catholic University of Eichstätt-Ingolstadt – Germany

Abstract

This article systematically reviews the literature on adaptive large neighborhood search (ALNS) to gain insights into the operators used for vehicle routing problems and their effectiveness. The ALNS has been successfully applied to a variety of optimization problems, particularly variants of the vehicle routing problem. The basic idea of the ALNS is to gradually improve an initial solution by modifying it using removal and insertion operators. However, relying solely on adaptive operator selection is not advisable. Instead, authors often conduct experiments to identify operators that improve the solution quality or remove detrimental ones. This process is mostly cumbersome due to the wide variety of operators, further complicated by inconsistent nomenclature. The goals of this article are to classify operators using consistent terminology, analyze their performance and establish a common basis for future research. To achieve these goals, we conduct a network meta-analysis of 211 articles that meet our criteria, and we employ incomplete pairwise comparison matrices, similar to rankings used in sports, to rank the operators. We identify 57 distinct removal and 42 insertion operators, and our analysis ranks them based on their effectiveness. Our findings reveal that sequence-based removal operators, which remove sequences of customers in the current solution, are the most effective. The best-performing insertion operators are those that exhibit foresight, such as regret insertion operators.
Finally, best practices and possible future research directions are discussed.

*Speaker

Hyperheuristic as Tuning Tool of Generalized Swap Strategy

Marek Kvet*[1] and Jaroslav Janáček

[1]University of Žilina – Slovakia

Abstract

This paper introduces a hyperheuristic, which is supposed to be used as a tuning tool of generalized swap strategy suggested for Pareto front approximation. Pareto front as a small subset of all feasible solutions is searched for whenever the solved location problem combines two contradictory criteria. The quality of obtained results will be evaluated based on comparing the complete Pareto front to its approximation. The used problem instances correspond to real systems established in self-governing regions of Slovakia.

*Speaker

Artificial Intelligence for Metaheuristic Parameter Setting

Jaroslav Janáček* and Marek Kvet*[1]

[1]University of Žilina – Slovakia

Abstract

The scientific content of this paper focuses on service system optimization. The main goal of presented research consists in extending current portfolio of solving approaches for such discrete location problems, in which two contradictory objectives are to be optimized. Due to different optimization criteria, a small subset of feasible solutions respecting a specific non-dominance property needs to be searched for. Obtaining the complete Pareto front is a time-consuming challenge. This phenomenon has led to the development of various metaheuristic approaches, which are able to bring an approximation of the original Pareto front. Mentioned approximate approaches may be sensitive to different parameters. That is why we pay attention to artificial intelligence for metaheuristic parameter settings. Suggested algorithm was tested on middle-sized benchmarks derived from real world. The obtained Pareto front approximations are compared to the complete sets of non-dominated solutions, which are available.

*Speaker

Trend-Risk in Complex Portfolio Selection Strategy

David Neděla*[1], Sergio Ortobelli[1,2], and Tomas Tichy[1]

[1]Technical University of Ostrava [Ostrava] – Czech Republic
[2]Università degli studi di Bergamo – Italy

Abstract

In this work, we provide an alternative trend (time)-dependent risk measure to Ruttiens' accrued returns variability (Ruttiens, 2013). We propose to adjust the calculation procedure to achieve an alternative risk measure. Our modification eliminates static mean component and it is based on the deviation of squared dispersions, which reflects the trend (time factor) precisely. Moreover, we also present a new perspective on dependency measures and we apply a PCA to a new correlation matrix in order to determine a parametric and nonparametric return approximation. In addition, the two-phase portfolio selection strategy is considered, where the mean–variance portfolio selection strategies represent the first optimization. The second one is the minimization of deviations from their trend leading to identical mean and final wealth. Finally, an empirical analysis verify the property and benefit of portfolio selection strategies based on these trend-dependent measures. In particular, the ex-post results show that applying the modified measure allows us to reduce the risk with respect to the trend of several portfolio strategies

*Speaker

Mixed Integer Linear Programming Based Large Neighborhood Search Approaches for the Directed Feedback Vertex Set Problem

Maria Bresich*[1], Johannes Varga[1], Günther R. Raidl[1], and Steffen Limmer[2]

[1]Institute of Logic and Computation [Vienna](TU Wien) – Austria
[2]Honda Research Institute Europe GmbH – Germany

Abstract

A directed feedback vertex set (DFVS) of a directed graph is a subset of vertices whose removal makes the graph acyclic. Finding a DFVS of minimum cardinality is the goal of the directed feedback vertex set problem, an NP-hard combinatorial optimization problem. We first consider two mixed integer linear programming (MILP) models for this problem, which, when solved with Gurobi, are effective on graphs of small to medium complexity but do not scale well to large instances. Aiming at better scalability and higher robustness over a large variety of graphs, we investigate a large neighborhood search (LNS) in which a destroy operator removes randomly chosen nodes from an incumbent DFVS and one of the MILP models is used for repair. Regarding the destroy operator, finding a best degree of destruction is challenging. A main contribution lies in proposing several selection strategies for this parameter as well as a strategy for choosing the more promising MILP model for repair. We evaluate the performance of the MILP models and different LNS variants on benchmark instances and compare the approaches to each other as well as to state-of-the-art procedures. Results show that our LNS variants yield clearly better solutions on average than standalone MILP solving. Even though our approaches cannot outperform the state-of-the-art, we gain valuable insights on beneficially configuring such a MILP-based LNS.

*Speaker