ATLANTIS PRESS

Journal of Robotics, Networking and Artificial Life
Vol. **6**(**1**); June (2019), *pp.* 56–59
DOI: https://doi.org/10.2991/jrnal.k.190602.001; ISSN 2405-9021; eISSN 2352-6386
https://www.atlantis-press.com/journals/jrnal

Research Article

Unified Approach to (1 + 1) Evolutionary Algorithm on Discrete Linear Functions

Kenji Aoki[1,*], Makoto Sakamoto[2], Hiroshi Furutani[3], Satoru Hiwa[4], Tomoyuki Hiroyasu[4]

[1]*Information Technology Center, University of Miyazaki, 1-1 Gakuen-kibanadai-Nishi, Miyazaki 889-2192, Japan*
[2]*Department of Computer Science and Systems Engineering, Faculty of Engineering, University of Miyazaki, 1-1 Gakuen-kibanadai-Nishi, Miyazaki 889-2192, Japan*
[3]*Innovative Computing Research Center, Doshisha University, 1-3 Tataramiyakodani, Kyoto 610-0394, Japan*
[4]*Department of Biomedical Information, Faculty of Life and Medical Sciences, Doshisha University, 1-3 Tataramiyakodani, Kyoto 610-0394, Japan*

ARTICLE INFO

Article History
Received 13 October 2018
Accepted 24 November 2018

Keywords
Evolutionary algorithm
discrete linear function
(1 + 1) EA
PO-mutation
PO-EA

ABSTRACT

We consider the runtime property of discrete linear functions in (1 + 1) Evolutionary Algorithms (EAs), Partially Ordered mutation (PO-mutation) and Jansen's model, Partially Ordered Evolutionary Algorithm (PO-EA). We analyze the process of evolution. This study was motivated by the paper of Jansen treating the runtime property of (1 + 1) EA on monotonic functions by means of probabilistic theory. As linear functions are special cases of monotonic function, we analyze their behavior. We show that the (1 + 1) EA can obtain an optimum solution at a stable hitting time for discrete linear functions. When the mutation rate is weak, on the order of $1/l$, most monotonic functions behave similarly and can be approximated well by the PO-mutation model.

1. INTRODUCTION

Discrete optimization is a generic term for mathematical programming that minimize or maximize a given function $f(x)$ under the constraint that the solution x belongs to a set feasible region with discrete properties. It is depending on whether the condition that defines the set feasible area is a combinational condition or an integer condition, it may be roughly divided into combinatorial optimization and integer optimization. In this research area, there are researchers who try to capture the characteristics of the optimization function $f(x)$ to make optimization easy to handle [1]. We are also interested in the properties of this optimization function. In the area of discrete optimization, there are many interesting properties that facilitate the optimization task [2]. One of them is monotony. Monotonic submodular functions have recently been applied in the field of machine learning. This paper focuses on optimization functions that are monotonic functions.

The Partial Order Evolutionary Algorithm (PO-EA) model introduced by Jansen [3] contributes to the analysis of performance in linear functions and is expected to simulate the optimization of monotonic functions [4,5]. PO-EA is a pessimistic model of the true optimization process, which can be used to derive an upper bound on the expected hitting time of monotonic functions. PO-EA was divided into two parts by Ma et al. [6]. These are called PO-mutation and ZeroMax models, respectively. In this study, we used a PO-mutation model. The PO-mutation model contains mutations that increase the fitness of every monotonic function.

In this study, we assumed that the optimum binary string is $\{1\}^l$, and use typical monotonous discrete linear functions on (1 + 1) EA, PO-mutation and PO-EA. The optimum hitting time calculated by computer experiments could confirm the contribution of (1 + 1) EA, PO-mutation and PO-EA, and help to understand the mechanism of the optimization process of EA.

2. MATHEMATICAL MODELS OF EAs

An EA is a generic term for a search model or algorithm for searching for an optimal solution by imitating the process of inheritance or evolution of an organism. This method evolves a group of solutions and obtains an optimal solution by repeating generational change based on change and selection on a problem. We can apply to various problems as long as we can represent gene numbers and set fitness evaluation functions. However, even in the case where a single search can be performed in a short time, a problem in which the number of all solutions is enormous may take too much time for the calculation itself and may not reach an effective solution. EA can be classified into genetic algorithm, evolutionary strategy, and evolutionary programming.

In this study, we use the simplest (1 + 1) EA, in which both the number of parent and offspring genes are one [7]. Genes are represented by binary string. A gene of length i can be written as $(x_1, x_2, ..., x_i)$ (where $x_i = 0$ or 1). In addition, gene mutation means that the value of each x_i is inverted according to the probability p_m.

[*] *Corresponding author. Email: aoki@cc.miyazaki-u.ac.jp*

We called it mutation probability. If the function $f(x)$ is defined in an interval $[a, b]$ and $f(x) \leq f(x')$ for any x, x' $(x < x')$ in the interval, $f(x)$ is said to be a monotonically increasing function in the interval $[a, b]$. If $f(x) \geq f(x')$, then $f(x)$ is said to be a monotonically decreasing function in the interval $[a, b]$. These are generically called monotonic function.

In the past research, Furutani et al. [8] analyzed the expected hitting times of EAs and Random Local Search by applying Markov chain. They showed that using $p_m = 1/l$, the runtime of the Markov chain is approximately given by $l \log(l)$, where l is the length of the gene and l is large enough. In this study, we also set $p_m = 1/l$.

3. EVOLUTIONARY ALGORITHMS

3.1. (1 + 1) Evolutionary Algorithm

$(\mu + \lambda)$ Evolutionary algorithm came from Evolutionary Strategy developed by Rechenberg and Schwefel, where μ and λ are numbers of parent and offspring solutions, respectively. We set $\mu = 1$ and $\lambda = 1$. These are the simplest combinations, and represented by (1 + 1) EA. It is too simple for an analysis, it has following properties [2].

(1) It is efficient for many problems.
(2) It cannot get stuck in a local optimum.
(3) The analysis of it reveals many tools that can be used in more practical algorithms.

The algorithm of (1 + 1) EA is given by Algorithm 1. In this algorithm, $\boldsymbol{x}$ are binary strings and $\boldsymbol{f}(\boldsymbol{x})$ are optimization functions.

Algorithm 1 | (1 + 1) EA

```
1: Initialize $x \in \{0, 1\}^l$ uniformly at random.
2: Create $x'$ by flipping one each bit in $x$ with probability $p_m$.
3: Select if $f(x') \geq f(x)$ then $x := x'$.
4: Go to 2 until a termination condition is fulfilled.
```

3.2. PO-mutation Algorithm

Colin et al. [5] have shown that PO-EA can be divided into two parts. Ma et al. [6] have named it PO-mutation and ZeroMax models. They treated two selection conditions, $(x' \geq x)$ and $\{(x' \nleq x) \text{ AND } (f(x') \leq f(x))\}$, separately. They called the first condition as PO-mutation model, and the second condition as ZeroMax model, respectively. PO-mutation has a role of driving force for approaching the optimum solution. In contrast, ZeroMax model works as a resistance against the approaching to the optimum solution. ZeroMax model's power is smaller than OneMax model because it has a condition $(x' \nleq x)$. The algorithm of PO-mutation is given by Algorithm 2.

Algorithm 2 | PO-mutation

```
1: Initialize $x \in \{0, 1\}^l$ uniformly at random.
2: Create $x'$ by flipping one each bit in $x$ with probability $p_m$.
3: Select if $x' \geq x$ then $x := x'$.
4: Go to 2 until a termination condition is fulfilled.
```

In the comparison x and x', $(x' > x)$ means that every bit changes were only from 0 to 1, $(x' < x)$ means that every bit changes were only from 1 to 0, and $(x' \nleq x)$ means that both bit changes were mixed.

We assumed that $\{1\}^l$ is the optimum string in the step 3 Selection process of Algorithm 2. PO-mutation model is used to investigate PO-EA convergence properties, so this model cannot be used to the problems whose the optimum string is not known.

3.3. Partial Order Evolutionary Algorithm

Jansen introduced the PO-EA model [3]. The algorithm of PO-EA is given by Algorithm 3.

Algorithm 3 | PO-EA

```
1: Initialize $x \in \{0, 1\}^l$ uniformly at random.
2: Create $x'$ by flipping one each bit in $x$ with probability $p_m$.
3: Select if $(x' \geq x)$ OR $\{(x' \nleq x)$ AND $(f(x') \leq f(x))\}$ then $x := x'$.
4: Go to 2 until a termination condition is fulfilled.
```

The difference of three algorithms are Selection in the step 3.

4. NUMERICAL EXPERIMENT

As an optimization function, we apply discrete linear function $f(x)$

$$
\begin{aligned}
f(x) &= a_1x_1 + a_2x_2 + \cdots + a_ix_i \\
f(x) &= \sum_{i=1}^{l} a_ix_i, \quad x_i \in \{0,1\}, \\
a_i &= (1+s)^i, \quad 0 \leq s \leq 1,
\end{aligned}
\tag{1}
$$

where x is a binary string of length l. We consider the maximization problem of this function. The optimum solution is $x_{\text{opt}} = \{1\}^l$. If $s = 0$, then $a = 1$ and $f(x) = \sum_{i=1}^{l} x_i$. That is to say OneMax function. If $s = 1$, then $a_i = 2^i$, i.e. binary number.

Since many studies suggested that the mutation probability of $p_m = 1/l$ may be the best choice, we carried out our analysis using this value.

In this section, we compare the first hitting time of the numerical experiments of (1 + 1) EA, PO-mutation and PO-EA with $s = 0.0$, 1.0 and 0.5. We used the mutation rate $p_m = 1/l$. The length of the string is $l = 100$, and we performed 10,000 runs for each parameter set, and averaged over them.

Figure 1 shows the time dependence of the probability of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with $s = 0.0$. These lines are the results of numerical calculation. The result is moving averaged with window size of 30. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation. In Figure 1, the waveforms of (1 + 1) EA and PO-mutation are similar, but PO-EA is behind the other. The mean of the first fitting times in (1 + 1) EA, PO-mutation and PO-EA are 1069, 1014 and 1976, respectively.

Figure 2 shows the time dependence of the probability of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with s = 1.0. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation. In Figure 2, the peak of the first hitting time is delayed in order of PO-mutation, (1 + 1) EA and PO-EA. The mean of the first fitting times in (1 + 1) EA, PO-mutation and PO-EA are 1136, 1008 and 1295, respectively.

Figure 3 shows the time dependence of the probability of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with s = 0.5. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation. In Figure 3, as in Figure 1, the waveforms of (1 + 1) EA and PO-mutation are similar. PO-EA is behind the other, but it is not as late as Figure 1. The mean of the first fitting times in (1 + 1) EA, PO-mutation and PO-EA are 1050, 1018 and 1280, respectively.

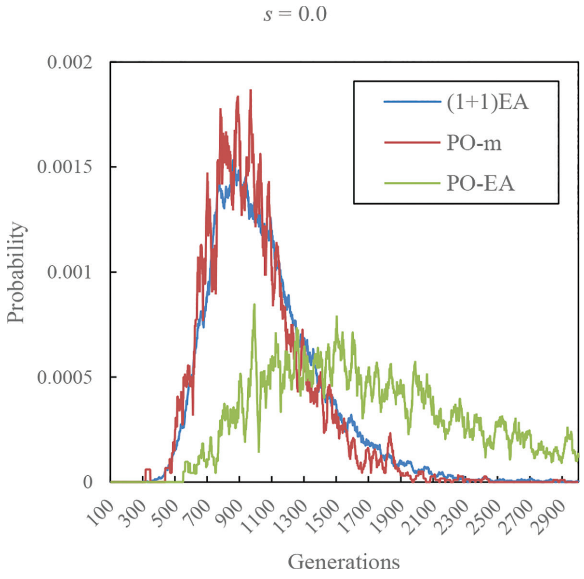

Figure 1 | The distribution of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with s = 0.0. The blue line is the result of (1 + 1) EA calculation. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation.

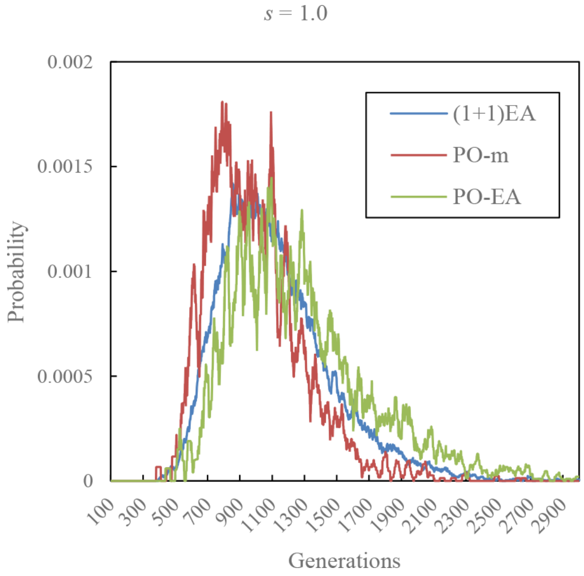

Figure 2 | The distribution of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with s = 1.0. The blue line is the result of (1 + 1) EA calculation. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation.

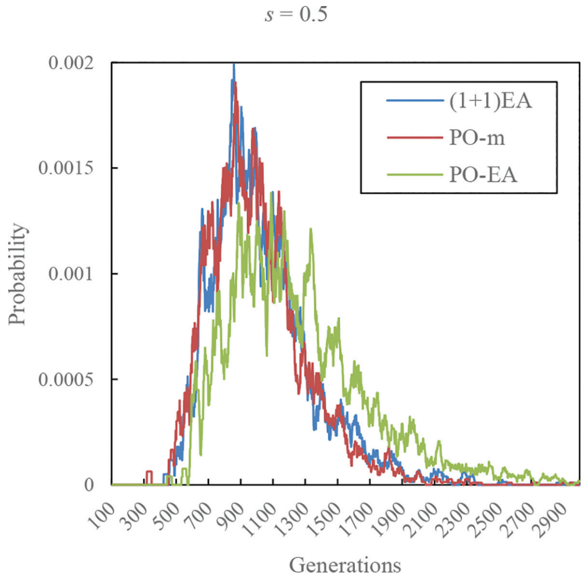

Figure 3 | The distribution of the first hitting time of optimum solution in (1 + 1) EA, PO-mutation and PO-EA with s = 0.5. The blue line is the result of (1 + 1) EA calculation. The red line is the result of PO-mutation calculation. The green line is the result of PO-EA calculation.

5. SUMMARY

In this paper, we studied time behavior of (1 + 1) EA, PO-mutation and PO-EA to understand how evolutionary computation works for discrete linear functions. We demonstrate that the (1 + 1) EA can obtain an optimum solution at a stable hitting time for discrete linear functions. When the mutation rate is weak, on the order of $1/l$, most monotonic functions behave similarly and can be approximated well by the PO-mutation model. These studies can help to understand the working mechanism of EA, and give some suggestions to design algorithms for other problems.

CONFLICTS OF INTEREST

There is no conflicts of interest.

REFERENCES

[1] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[2] S. Fujishige, Submodular Functions and Optimization, second ed., Elsevier, Amsterdam, 2005.

[3] T. Jansen, On the brittleness of evolutionary algorithms, in: C.R. Stephens, M. Toussaint, D. Whitley, P.F. Stadler, (Eds.),

Foundations of Genetic Algorithms, (FOGA 2007), Lecture Notes in Computer Science, Vol. 4436, Springer, Heidelberg, 2007, pp. 54–69.

[4] B. Doerr, T. Jansen, D. Sudholt, C. Winzen, C. Zarges, Mutation rate matters even when optimizing monotonic functions, Evol. Comput. 21 (2013), 1–27.

[5] S. Colin, B. Doerr, G. Férey, Monotonic functions in EC: anything but monotone!, Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada, 2014, pp. 753–760.

[6] Q.L. Ma, Y-a. Zhang, K. Yamamori, H. Furutani, S. Hiwa, T. Hiroyasu, Markov chain analysis of evolutionary algorithms for monotonic functions, Artif. Life Robot. 24 (2019), 82–87.

[7] P.S. Oliveto, X. Yao, Runtime analysis of evolutionary algorithms for discrete optimization, Chapter 2, Theory of Randomized Search Heuristics, World Scientific, Singapore, 2011, pp. 21–52.

[8] H. Furutani, H. Tagami, M. Sakamoto, Y. Du, Markov chain analyses of random local search and evolutionary algorithm. J. Robot. Netw. Artif. Life. 1 (2014), 220–224.

Authors Introduction

Dr. Kenji Aoki

He received his PhD of Engineering from Kagoshima University in 2010. He is currently working in Information Technology Center at University of Miyazaki as Associate Professor, since 2010. His research interests include bio-informatics, evolutionary computation, information system and Intelligent systems. He is a member of IPSJ and JSET.

Dr. Makoto Sakamoto

He received his PhD degree in computer science and systems engineering from Yamaguchi University in 1999. He is presently an associate professor in the Faculty of Engineering, University of Miyazaki. His first interests lay in hydrodynamics and time series analysis, especially the directional wave spectrum. He is a theoretical computer scientist, and his current main research interests are automata theory, languages and computation. His articles have appeared in journals such as Information Sciences, Pattern Recognition and Artificial Intelligence, WSEAS, AROB, ICAROB, SJI, IEEE and IEICE (Japan), and proceedings of conferences such as Parallel Image Processing and Analysis, SCI, WSEAS, AROB and ICAROB.

Dr. Hiroshi Furutani

He received his PhD of Science from Kyoto University in 1981. He is currently working in Innovative Computing Research Center at Doshisha University as researcher. His research interest includes bio-informatics, evolutionary computation and quantum computing.

Dr. Satoru Hiwa

He received his PhD in Engineering from Doshisha University. He is currently working in Faculty of Life and Medical Sciences at Doshisha University, Kyoto, Japan, as an Associate Professor. His current research interests include human brain mapping, non-invasive neuroimaging and computational intelligence.

Dr. Tomoyuki Hiroyasu

He received his doctor of Engineering from Waseda University in 1998. He has been a Professor of Life and Medical Sciences, Doshisha University, since 2008. His research interests include the development of novel evolutionary algorithms, medical image processing, medical information systems, parallel computing, Grid, web systems, and Intelligent systems. He is a member of IEEE, JSME, JSCES, IPSJ, SICE, and IEICE.