Applied Soft Computing 13 (2013) 4536–4547

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# A block-based evolutionary algorithm for flow-shop scheduling problem

Pei-Chann Chang[a,*], Meng-Hui Chen[a], Manoj K. Tiwari[b], Asif Sikandar Iquebal[b]

[a] *Department of Information Management, Yuan Ze University, 135 Yuan Tung Road, Chung-Li, Taoyuan 32026, Taiwan*
[b] *Department of Industrial Engineering, Indian Institute of Technology, Kharagpur 721302, West Bengal, India*

ARTICLE INFO

*Article history:*
Received 25 August 2012
Received in revised form 23 May 2013
Accepted 16 July 2013
Available online 4 September 2013

*Keywords:*
Evolutionary algorithm
Gene linkage
Combinatorial optimization problem
Building blocks
Recombination

ABSTRACT

Combinatorial problems like flow shop scheduling, travel salesman problem etc. get complicated and are difficult to solve when the problem size increases. To overcome this problem, we present a block-based evolutionary algorithm (BBEA) which will conduct evolutionary operations on a set of blocks instead of genes. BBEA includes the block mining and block recombination approaches. A block mining algorithm is developed to decompose a chromosome into a set of blocks and rest of genes. The block is with a fixed length and can be treated as a building block in forming a new chromosome later on. To guide the block mining process, a gene linkage probability matrix is defined that shows the linkage strength among genes. Therefore the blocks can be further evolved during the evolutionary processes using this matrix. In the block recombination approach, the blocks along with the rest of genes are recombined to form a new chromosome. This new evolutionary approach of BBEA is tested on a set of discrete problems. Experimental results show that BBEA is very competitive when compared with traditional GA, EA or ACGA and HGIA approaches and it can largely improve the performance of evolutionary algorithm and save a fair amount of computational times simultaneously.

## 1. Introduction

Almost all the genetic algorithms (GAs) rely on genetic operators such as selection, crossover and mutation to search the solution spaces for optimal solutions using some kind of fitness values. GAs are powerful search techniques that are used successfully to solve problems in many different disciplines. However, for most combinatorial problems such as traveling salesman problems (TSP), machine scheduling problems, and vehicles routing problems are very difficult to solve and even for moderate cases the GA converges prematurely to some local optima owing to the computational complexities of the problem.

The basic issues of the current developments in genetic algorithm [1] include the following: (1) Chromosome representation: When compared with the DNA or RNA structures, the chromosome representation of most problems are not sufficient enough to show the sophisticated structure of natural representation of DNA or RNA and there are still lots of rooms for improvement in this chromosome representation. (2) Design of operators: Operators such as crossover or mutation involving the exchange of a set of genes within the chromosome are not very efficient and effective and they may produce lots of redundant moves during the evolutionary processes. For example in case of flow shop scheduling problem, if the sequence is disturbed even slightly more than required, then it may completely mislead the solution. In some other cases, the genetic operators will be very inefficient in dealing with large size of combinatorial problems. (3) Selection: Natural selection is much diversified when compared with the current selection methods such as roulette wheel or tournament selections. Looking towards the issues of simple genetic algorithm, this research attempts to develop a novel approach by providing a block-based evolutionary approach which is more efficient in dealing with the combinatorial optimization problems.

The research is based on the concept of schema theorem to locate blocks from a set of chromosomes with good fitness. These short, low-order and highly fit schemata are called blocks. It is not always possible to get good blocks every time as the quality of the blocks depend on the problem as well as the chromosomes generated in previous generations. Therefore, this research is quite different from the previous researches which try to learn linkage relationships or building blocks (BBs) to form a new method [2–4]. Instead, the aim of this work is to come out with a novel approach which can locate a block, i.e., a series of genes, from a set of high-quality chromosomes or elites. As we move towards the end of the evolution process, the quality of chromosomes increases, hence better blocks are generated which leads to better solutions. We then recombine these blocks mined from the elite to generate a new chromosome. Toward this goal, a simplified definition of BBs

---

* Corresponding author. Tel.: +886 936 101320; fax: +886 346 38884.
*E-mail address:* iepchang@saturn.yzu.edu.tw (P.-C. Chang).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.07.018

is proposed. Blocks are similar contiguous bits found in highly fit chromosomes. The motivation of this research is to merge these blocks with other genes to form a new chromosome. Consequently, the resulting chromosomes are of higher quality.

The paper is organized as follows. The next section provides a literature reviews on BBs in GA. Section 3 introduces the block mining and recombination approach for flow shop scheduling problems. Section 4 validates the algorithm with experimental results. Finally, Section 5 gives discussion and conclusion.

## 2. Literature reviews

The BBs are common structures of high-quality solutions. It is assumed that high-quality solutions can be obtained by combining the BBs. One obvious solution is finding a way to identify better BBs explicitly and then recombine them in an appropriate manner to form a legal chromosome. The BBs can be regarded as the linkage between two or more alleles [5]. There are many ways to determine the linkage association such as loosely or tightly. A model of the linkage can be built in several manners and can be identified explicitly. In general, the meaning of the linkage model is equivalent to the BBs.

In order to prevent the early convergence in GA as mentioned by Holland [1], and to assure the success of genetic algorithm, it is important to mine linkage information to recombine alleles. A GA approach was proposed by Goldberg et al. [6] that proper problem decomposition is one of the keys to ensure the effectiveness of GA. An approach that concerns with explicit BBs is the messy GA (mGA) [7]. The mGA allows schema redundancy, and uses cut and splice technique as recombination operators. The mGA's mechanism and its BBs outperform the simple GAs (sGA) in many problems. Later, the mGA is improved in various versions [8,9]. Since then, many researches in linkage learning design for GA [10] have been developed. The linkage learning GA (LLGA) [11–13] that the chromosome is represented as a circular structure and the probabilistic expression mechanism is used for interpreting the chromosome. The recombination process uses the exchange crossover which performs linkage skew and linkage shift. Performance of the LLGA is superior to the simple GA on exponentially-scaled problems. Different linkage learning methods have been explained like explicit linkage learning, e.g., linkage identification by nonlinearity check procedure (LINC) [14], probabilistic model building GAs (PMBGA) [15], or estimation of distribution algorithms (EDA) [16]. The deterministic e.g., dependency structure matrix GA (DSMGA) [17]. However, the problem with explicit linkage learning is that the probabilistic model also prematurely gets trapped into local optimum. As for the explicit model, it is hard to identify a set of blocks or linkage set from the chromosome in real world application.

A very similar but different to mGA is proposed in [3]. The authors present a novel co-evolutionary algorithm, the Puzzle Algorithm, where a population of BBs coevolves alongside a population of solutions. They show that the addition of a building-block population to a standard evolutionary algorithm results in notably improved performance on the hard shortest common superstring (SCS) problem. However, due to the nature of BBs that are dependent on the problems and the encoding of the chromosome, their behaviors are difficult to analyze. General problem-independent GAs such as mGA [8] and the puzzle algorithm [3] are not very efficient in solving TSPs, especially for large problems.

A lot of work has been done in the past to determine the linkage sets. Most of them can be categorized into two classes. One is called perturbation methods (PMs) and the other is called estimation of distribution algorithms (EDAs). PMs perturb values (or combinations of values) and calculate fitness differences to obtain linkage information. EDA replaces traditional crossover and mutation of GAs by (1) building a probabilistic model of promising strings and (2) sampling the built model to generate new strings. Advanced EDAs involve conditional or marginal probabilities to encode relationships between variables. For example, extended compact genetic algorithm (ECGA) [16] detects sets of variables tightly linked to form a building block by computing a combined complexity measure. The combined complexity is the sum of model complexity and compressed population complexity. The ECGA, which is more GA-like algorithm than other EDAs, combines strings according to the obtained sets of variables and do not employ the probabilistic sampling which most of EDAs do employ. The schema theorem assumes a positive effect of the selection that can maintain the good schema, and shows the negative effect of the crossover and the mutation that disrupt the good schema. However there is no guideline how to process the BBs and the analysis is limited to the progress made in one generation. There is an effort to measure quantity of the BBs [18]. Many problems are analyzed: OneMax, Trap, Parabola and TSP problem. Two encoding schemes are used: the binary encoding and the gray encoding. The results show that BBs exist in OneMax, Trap, Parabola (the gray coding) and TSP (with third encoding scheme: binary matrix). This shows that the BBs existence also depend on the encoding scheme. There are many factors that affect BBs such as the selection method, the identification algorithm, the recombination procedure and the measurement criterion.

In the earlier researches [19–21], ACGA has shown up as a very successful method in injecting ACs into the evolutionary process of GA to speed up the convergence. However, the solution quality still can be further improved when compared with other approaches. In this research, a block-based genetic algorithm is introduced. BBEA contains several special characteristics. Firstly, the new method uses gene linkage probability matrix update rules to extract the blocks from the population generated in the previous generations. It is a process of linkage learning which is applied to discover the hidden knowledge within the dependent variables. The good substructures can be interpreted as common subsequences among a set of highly fit chromosomes. The block consists of a series of genes linked to each other continuously. Secondly, a recombination procedure is adopted to regroup the blocks and the rest of jobs together to generate a new chromosome. These new generated chromosomes are built with very good subsequences (or, micro-structures) and they will be recombined to form a new chromosome. The block mining by gene linkage probability matrix update is very effective which can identify a good subsequence from the chromosome. These blocks can reduce the size of the search space, so that the search process would take less time to find the near-optimal solution. In summary, BBEA is introduced to further enhance the search effectiveness and efficiency.

## 3. A block-based evolutionary algorithm (BBEA)

The framework of the BBEA is shown in Fig. 1. The key processes include block mining and recombination operator. We will introduce the block mining first and then the recombination procedure.

In most applications, GAs can search excellent solutions even the optimal solutions. However, in the applications of high complexities, GA is usually trapped in the local optima and hard to reach the global optima. BBEA aims at mining the linkage information of genes to assemble the strong joining blocks and recombine the mined blocks with rest genes to generate ACs to enhance the solutions structure. Due to the composition of solutions which contains the different permutations with the natural evolved solutions; the BBEA has the ability to extend the searching space and to locate more promising solution when compared with other meta-heuristics.