



A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem

Pei-Chann Chang^a, Wei-Hsiu Huang^a, Jheng-Long Wu^{a,*}, T.C.E. Cheng^b

^a Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, ROC

^b Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:

Received 11 February 2012

Accepted 4 June 2012

Available online 15 June 2012

Keywords:

Building blocks

Genetic algorithm

Artificial chromosomes

Linkage learning

Re-combination

Flowshop scheduling

ABSTRACT

The goal of block mining is to obtain a set of genes that contain dependency among gene relationships. Such blocks without overlapping of genes can be further merged to form a new chromosome and the quality of the new chromosome can be greatly improved. Based on this concept, we propose a novel block mining method that is able to locate common structures or to establish new blocks (like a small piece of puzzle) from a set of high fit chromosomes. The identified blocks (puzzles) will also be updated generation by generation through the newly updated high fit chromosomes. We develop a heuristic re-combination procedure to form a new chromosome by re-combining the blocks. We call the new chromosomes generated as artificial chromosomes (ACs) and inject them into the evolutionary process when the convergence slope of the evolutionary process is less than a predefined threshold. This new algorithm retains the regular simple genetic algorithm (SGA) operations of crossover and mutation, and utilizes the ACs generated from elites to speed up the convergence process. Experimental results indicate that the puzzle-based method of chromosome generation is very efficient and effective in solving the traditional permutation flowshop scheduling problem. The new method can be applied to tackle other NP-complete problems such as scheduling and vehicle routing problems.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Genetic algorithm (GAs) relies on genetic operators such as selection, crossover, and mutation to search the solution space for optimal solutions. GA is a powerful search technique, which has been successfully applied to solve combinatorial problems in different disciplines. However, for combinatorial problems that are very difficult to solve even in moderate sizes such as the traveling salesman problem (TSP), the vehicle routing problem (VRP), and various machine scheduling problems, GA will converge prematurely. In other words, GA will be trapped into a local optimum at an early stage and cannot deliver the global optimal solution.

To prevent early convergence of GA as observed by Holland (1992), it is important to acquire linkage information to recombine building blocks to ensure the success of GA. In addition, Goldberg et al. (1992) suggest that proper problem decomposition is one of the keys to ensure the effectiveness of GA. Since then, much research on the linkage learning design for GA has been conducted. The linkage model can be implicit, e.g., linkage

learning GA (LLGA) (Harik and Goldberg, 1996), explicit, e.g., linkage identification by the nonlinearity check procedure (LINC) (Munetomo and Goldberg, 1999), probabilistic model building GA (PMBGA) (Pelikan et al., 2002), estimation of distribution algorithm (EDA) (Larrañaga and Lozano, 2001), or deterministic, e.g., dependency structure matrix GA (DSMGA) (Yu et al., 2003). However, the problem with explicit linkage learning is that the probabilistic model will also be premature and trapped into a 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 applications.

This research is based on the concept of the schema theorem to locate blocks from a set of high fit chromosomes. These short, low-order, and highly fit schemata are called “blocks”. Blocks can be considered as a plausible explanation for the success of GA. However, due to the nature of blocks, which depends on the problem and encoding of the chromosome, their behaviors are difficult to analyze. Therefore this research is different from previous studies that try to learn linkage relationships or building blocks (BBs) to create a new method. Instead, we propose a novel approach that locates a series of genes physically from a set of high-quality chromosomes or elites. We then apply the blocks mined from the elites to generate a set of artificial chromosomes, which are then injected into the evolutionary process to speed up the convergence of SGA. To reach this goal, we propose

* Corresponding author.

E-mail addresses: iepchang@saturn.yzu.edu.tw (P.-C. Chang), wei-hsiu@saturn.yzu.edu.tw (W.-H. Huang), jlwu.yzu@gmail.com (J.-L. Wu), Edwin.Cheng@inet.polyu.edu.hk (T.C.E. Cheng).

a simplified definition of building blocks, called blocks. Blocks are similar contiguous bits found in highly fit chromosomes. The motivation for this research is to merge such blocks with other genes to form a new chromosome. The quality of the new chromosome will be greatly improved.

The paper is organized as follows: Section 2 provides a literature review on the concept of building blocks and on the flowshop scheduling problem. Section 3 introduces the block mining and re-combination approach, and applies it to solve the flowshop scheduling problem. Section 4 presents the experimental results to evaluate the effectiveness of the proposed algorithm. Finally, Section 5 discusses the results, concludes the paper, and suggests topics for future research.

2. Literature review

We review the related literature on building blocks and the flowshop scheduling problem in this section.

2.1. Building blocks

Building blocks are common structures of high-quality solutions. It is assumed that high-quality solutions are composed of building blocks and the solution quality can be improved by composing building blocks. One obvious solution is to find a way to identify BBs explicitly and then re-combine them to form a legal chromosome. BBs can be regarded as the linkage between two or more alleles (Chen et al., 2007). There are many ways to determine the linkage association such as loose or tight. 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 BBs.

There have been many efforts to detect linkage sets. Most of them can be categorized into two classes. One is called the perturbation method (PM) and the other is called the estimation of distribution algorithm (EDA). PM perturbs values (or combinations of values) and calculates the fitness differences to obtain linkage information. EDA replaces the traditional crossover and mutation operators of GAs by (1) building a probabilistic model of promising strings and (2) sampling the built model to generate new strings. Advanced EDA involves conditional or marginal probabilities to encode relationships between variables. For example, extended compact genetic algorithm (ECGA) (Larrañaga and Lozano, 2001) 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. ECGA, which is more GA-like algorithm than other EDAs, combines strings according to the obtained sets of variables and does not employ probabilistic sampling that most EDAs do. The schema theorem assumes a positive effect of the selection that can maintain a good schema and shows the negative effects of crossover and mutation that disrupt a good schema. However there is no guideline on how to process BBs and the analysis is limited to the progress made in one generation. There is an effort to measure the quantity of BBs by Aporntewan and Chongstivvatana (2005). Many problems are analyzed, including OneMax, Trap, Parabola and TSP. Two encoding schemes are used: binary encoding and gray encoding. The results show that BBs exist in OneMax, Trap, Parabola (gray coding) and TSP (with a third encoding scheme—binary matrix). This implies that the existence of BBs also depends on the encoding scheme. There are many factors that affect BBs such as the selection method, the identification algorithm, the re-combination procedure, and the measurement criterion.

An approach that concerns explicit BBs is the messy GA (mGA) (Goldberg et al., 1989). The mGA allows schema redundancy, and uses the cut and splice techniques as re-combination operators. The mGA's mechanism and its BBs outperform simple GAs (sGA) in many problems. Later, mGA is improved in various versions (Goldberg et al., 1993; Kargupta, 1996). Another concept is the linkage learning genetic algorithm (LLGA) (Harik and Goldberg, 1996, 2000; Harik et al., 1999). For LLGA, the chromosome is represented as a circular structure and the probabilistic expression mechanism is used for interpreting the chromosome. The re-combination process uses exchange crossover that performs linkage skew and linkage shift. The performance of LLGA is superior to SGA in treating exponentially-scaled problems.

2.2. The flowshop scheduling problem

The earliest GA for the flowshop scheduling problem (PFSP) was proposed by Chen et al. (1995). They generate the initial population by using several heuristic rules. In their approach, partially mapped crossover is applied, and there is no mutation. Reeves (1995) also propose a GA using the NEH heuristic as the initial population. The algorithm uses one-point order crossover and adaptive mutation. A GA with the elitism strategy was proposed by Murata et al. (1996). Their approach includes two-point order crossover, a shift mutation, and combination with local search. Reeves and Yamada (1998) propose a special crossover of GA, called multi-step crossover. Ponnambalam et al. (2001) propose a GA with generalized position crossover, a shift mutation, and random population initialization. Ruiz et al. (2006) propose two new advanced GAs. The initialization of the population is generated using heuristic methods. The algorithm uses four new crossover operators and a restart scheme that restarts a given portion of the population if the solution has not been improved for a given number of generations. More recently, Nagano et al. (2008) propose a constructive genetic algorithm (CGA). Their approach includes a population of dynamic size of schemata and structures, and the possibility of using heuristics in structure representation and in the fitness function definitions.

Some researchers try to combine GA with EDA. Pena et al. (2004) propose a hybrid algorithm that mixes the generic chromosomes and artificial chromosomes generated by EDA. Zhang et al. (2009) propose a hybrid genetic algorithm for the PFSP. One of the chromosomes in the initial population is constructed by a suitable heuristic. An artificial chromosome is generated by a weighted simple mining gene structure. In addition, local search is used to improve all the generated chromosomes in each generation.

Our previous research on ACGA and self-guided GA falls in the category of hybrid algorithms involving GA and EDA, proposed by Chang et al. (2008a) and Chen et al. (2009), respectively. ACGA uses artificial chromosomes to speed up the convergence of the GA. The artificial chromosomes are generated according to a probability matrix that is transformed from the dominance matrix by mining the gene structure of an elite base. The self-guided GA estimates the quality of a candidate offspring based on the probabilistic model used in its crossover and mutation operations. In such a way, the mutation and crossover operations are able to generate better solutions, thus improving the performance of the algorithm.

3. Block mining and re-combination enhanced GA for PFSP

The PFSP to minimize the makespan can be defined as follows: suppose there are n jobs and m machines. Let $p(i; j)$, $1, \dots, i, \dots, n$; $1, \dots, j, \dots, m$, be the processing time of job i on machine j and $i=(1, \dots, n)$ be a job permutation (i.e., a processing order of the

Download English Version:

<https://daneshyari.com/en/article/5080711>

Download Persian Version:

<https://daneshyari.com/article/5080711>

[Daneshyari.com](https://daneshyari.com)