Neurocomputing 173 (2016) 864–874

Contents lists available at ScienceDirect

Neurocomputing

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

Discrete state transition algorithm for unconstrained integer optimization problems

Xiaojun Zhou [a,b,*], David Yang Gao [b], Chunhua Yang [a], Weihua Gui [a]

[a] School of Information Science and Engineering, Central South University, Changsha 410083, China
[b] School of Science, Information Technology and Engineering, Federation University Australia, Victoria 3353, Australia

ARTICLE INFO

Article history:
Received 9 March 2014
Received in revised form
8 June 2015
Accepted 17 August 2015
Communicated by W. Lu
Available online 28 August 2015

Keywords:
State transition algorithm
Integer optimization
Traveling salesman problem
Maximum cut problem
Discrete value selection

ABSTRACT

A recently new intelligent optimization algorithm called discrete state transition algorithm is considered in this study, for solving unconstrained integer optimization problems. Firstly, some key elements for discrete state transition algorithm are summarized to guide its well development. Several intelligent operators are designed for local exploitation and global exploration. Then, a dynamic adjustment strategy "risk and restoration in probability" is proposed to capture global solutions with high probability. Finally, numerical experiments are carried out to test the performance of the proposed algorithm compared with other heuristics, and they show that the similar intelligent operators can be applied to ranging from traveling salesman problem, boolean integer programming, to discrete value selection problem, which indicates the adaptability and flexibility of the proposed intelligent elements.

1. Introduction

In this paper, we consider the following unconstrained integer optimization problem

$$\min f(x), \tag{1}$$

where, $x = (x_1, \ldots, x_n) \in \mathbb{Z}^n$.

Generally speaking, the above optimization problem is NP-hard, which cannot be solved in polynomial time. A direct method is to adopt the so-called "divide-and-conquer" strategy, which separates the optimization problem into several subproblems and then solve these subproblems step by step. Branch and bound (B&B), branch and cut (B&C), and branch and price (B&P) belong to this kind; however, these methods are essentially in exponential time. An indirect method is to relax the optimization problem by loosening its integrality constraints to continuity and then solve the continuous relaxation problem or its Lagrangian dual problem, including LP-based relaxation, SDP-based relaxation, and Lagrangian relaxation. Nevertheless, when rounding off the relaxation solution, they may cause some infeasibility or can only get approximate solutions, and when using Lagrangian dual, there may exist duality gap between the primal and the dual problem [4,6,8,11].

On the other hand, some stochastic algorithms, such as genetic algorithm (GA) [1,21], simulated annealing (SA) [9,23], ant colony optimization (ACO) [3,15], are also widely used for integer optimization problems, which aim to obtain "good solutions" in reasonable time. In terms of the concepts of state and state transition, a new heuristic search algorithm called state transition algorithm (STA) has been proposed recently, which exhibits excellent global search ability in continuous function optimization [24–28]. In [20], three intelligent operators (geometrical operators) named swap, shift and symmetry have been designed for discrete STA to solve the traveling salesman problem (TSP), and it shows that the discrete STA outperforms its competitors with respect to both time complexity and search ability. In [29], a discrete state transition algorithm is successfully applied to the optimal design of water distribution networks. To better develop discrete STA for medium-size or large-size discrete optimization problems, in the study, we firstly build the framework of discrete state transition algorithm and propose five key elements for discrete STA, of which, the representation of a decision variable, the local and global operators and the dynamic adjustment strategy are mainly studied. Four geometrical operators named swap, shift, symmetry and substitute are designed, which are intelligent due to their adaptability and flexibility in various types of integer optimization. The mixed strategies of "greedy criterion" and "risk and restoration in probability" are proposed, in which, "greedy criterion" and "restoration in probability" are used to guarantee a good convergence

* Corresponding author. Tel.: +86 13787052648.
E-mail addresses: michael.x.zhou@csu.edu.cn (X. Zhou), d.gao@federation.edu.au (D.Y. Gao), ychh@csu.edu.cn (C. Yang), gwh@csu.edu.cn (W. Gui).

http://dx.doi.org/10.1016/j.neucom.2015.08.041
0925-2312/

performance, and “risk a bad solution in probability” aims to escape from local optimality. Some applications ranging from traveling salesman problem, boolean integer programming, to discrete value selection problem are studied. Experimental results have demonstrated the effectiveness and efficiency of the proposed method.

The main contribution and novelty of this paper is three-fold, which can be summarized as follows: (1) a systematic formulation of discrete state transition algorithm is firstly proposed, including the state space representation and five key elements; (2) a dynamic adjustment strategy called “risk and restoration in probability” is designed to improve the ability to escape from local optima; (3) the proposed algorithm is successfully integrated with several classical integer optimization problems.

2. The framework of discrete state transition algorithm

If a solution to a specific optimization problem is described as a state, then the transformation to update the solution becomes a state transition. Without loss of generality, the unified form of generation of solution in discrete state transition algorithm can be described as

$$\begin{cases} x_{k+1} = A_k(x_k) \oplus B_k(u_k) \\ y_{k+1} = f(x_{k+1}) \end{cases}, \tag{2}$$

where, $x_k \in \mathbb{Z}^n$ stands for a current state, corresponding to a solution of a specific optimization problem; u_k is a function of x_k and historical states; $A_k(\cdot)$, $B_k(\cdot)$ are transformation operators, which are usually state transition matrixes; $\oplus$ is a operation, which is admissible to operate on two states; f is the fitness function.

As an intelligent optimization algorithm, discrete state transition algorithm has the following five key elements:

(1) *Representation of a solution*: In discrete STA, we choose a special representation, that is, the permutation of the set $\{1, 2, \ldots, n\}$, which can be easily manipulated by some intelligent operators. The reason that we call the operators “intelligent” is due to their geometrical property (swap, shift, symmetry and substitute), and an intelligent operator has the same geometrical function for different types of problems. A big advantage of such a representation and operators is that, after each state transformation, the newly created state is always feasible, avoiding the trouble into rounding off a continuous solution into an integral one.

(2) *Sampling in a candidate set*: When a transformation operator is exerted on a current state, the next state is not deterministic, that is to say, there are possibly different choices for the next state. It is not difficult to imagine that all possible choices will constitute a candidate set, or a “neighborhood”. Then we execute several times of transformation, called search enforcement (*SE*) degree, on current state, to sample in the “neighborhood”. Sampling is a very important factor in state transition algorithm, which can characterize the search space and avoid enumeration.

(3) *Local exploitation and global exploration*: In continuous optimization, it is quite significant to design good local and global operators. The local exploitation can guarantee high precision of a solution and convergent performance of an algorithm, and the global exploration can avoid getting trapped into local minima or prevent premature convergence. In discrete optimization, it is extremely difficult to define a “good” local optimal solution due to its dependence on a problem's structure, which leads to the same difficulty in the definition of local exploitation and global exploration. Anyway, in discrete state transition algorithm, we define a little change to current solution by a transformation as local exploitation, while a big change to current solution by a transformation as global exploration.

(4) *Self-learning and regular communication*: State transition algorithm behaves in two styles, one is individual-based, the other is population-based, which is certainly an extended version. The individual-based state transition algorithm focuses on self-learning, in other words, it focuses on designing operators and dynamic adjustment (details given in the following). Undoubtedly, communication among different states is a promising strategy for state transition algorithm, as indicated in [26]. Through communication, states can share information and cooperate with each other. However, how to communicate and when to communicate are key issues. In continuous state transition algorithm, intermittent exchange strategy was proposed, which means that states communicate with each other at a certain frequency in a regular way.

(5) *Dynamic adjustment*: It is a potentially useful strategy for state transition algorithm. In the iterative process of searching, the fitness value can decrease sharply in the early stage, but it stagnates in the late stage, due to the static environment. As a result, some perturbation should be added to activate the environment. In fact, dynamic adjustment can be understood and implemented in various ways. For example, the alternative use of different local and global operators is a dynamic adjustment to some extent. Then, we can change the search enforcement degree, vary the fitness function, reduce the dimension, etc. Of course, “risk a bad solution in probability” is another dynamic adjustment, which is widely used in simulated annealing (SA). In SA, the Metropolis criterion [12] is used to accept a bad solution: $p = \exp(-\Delta E/k_B T)$, where, $\Delta E = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)$, k_B is the Boltzmann probability factor, T is the temperature to regulate the process of annealing. In the early stage, temperature is high, and it has big probability to accept a bad solution, while in the late stage, temperature is low, and it has very small probability to accept a bad solution, which is the key point to guarantee the convergence. We can see that the Metropolis criterion has the ability to escape from local optimality, but on the other hand, it will miss some “good solutions” as well.

In discrete STA, a novel strategy, named “risk and restoration in probability”, is proposed. Details can be found in the following individual-based STA.

2.1. Individual-based discrete STA

In this part, we focus on the individual-based discrete STA, and the main process of discrete STA is shown in the pseudocode as follows:

```
repeat
  [Best,fBest] ← swap(*,Best,fBest)
  [Best,fBest] ← shift(*,Best,fBest)
  [Best,fBest] ← symmetry(*,Best,fBest)
  [Best,fBest] ← substitute(*,Best,fBest)
  if fBest < fBest* then        ▹ greedy criterion
    Best* ← Best
    fBest* ← fBest
  end if
  if rand < p1 then        ▹ restoration in probability
    Best ← Best*
    fBest ← fBest*
  end if
until the specified termination criteria are met
```