Applying Modified Colonial Competitive Algorithm to Solve Minimal Hit ting Set Problems

Chuanjun Zhu[1], Jing Cao[1], Chaoyong Zhang[2], and Kunlei Lian[2]

[1]Hubei University of Technology, Wuhan, 430068

[2]State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science and Technology, Wuhan, 430074

Abstract

A CCA was developed and modified to solve the problem of minimal candidates set. The minimal candidate set was a minimal hitting set. The modified CCA improved performance in initialization, to the population of countries maintained by CCA by introducing a third type of country, independent country, to the population of countries maintained by CCA. Implementation details of the proposed CCA and modified colonial competitive algorithm (MCCA) were elaborated using an illustrative example. The performance of the algorithms was analyzed, and the results by the MCCA were compared with DMDSE-Tree algorithm. When 90% of the minimal hitting sets are obtained, the MCCA has better efficiency. Finally, the experimental results of certain system verify the effectiveness of the algorithm, which proves that this method can be applied in solving the minimal hitting set of combinatorial optimization problems for selection of equipment effectively.

Key words: business planning; colonial competitive algorithm (CCA); minimal hitting set; equipment selection

1 Introduction

There exist many real-world problems that can be modelled as identifying the minimal set from the non-empty intersection with each set in a given group. The minimal hitting set problem refers to such problems and examples include the creation of model-based diagnosis from the smallest conflict set, the search for gene fragment to expound certain gene characteristics in gene expression analysis and the decision problem when collective uses make book purchases at public expense. There have been lots of efforts in the research community to improve efficiency of solving the minimal hitting set problem, which is NP-hard. Reiter proposed the first method in calculating the minimal hitting set problem in model-based diagnosis using HS-Tree, but it suffers from loss of feasible solutions due to its adoption of pruning and closing strategies in its search process. Greiner proposed an improved HS-Tree algorithm named HS-DAG that uses the idea of acyclic graphs in order to identify all the minimal hitting sets. The minimal hitting set problem has received lots of attentions from research community and many algorithms are proposed to solve this problem more effectively, which can be classified into exact algorithms and heuristic algorithms. Examples of exact algorithms include the algorithm based on BNB-HSSE, hybrid algorithm based on CHS tree and recursive boolean algorithm, hybrid algorithm based on HSSE tree and binary mark, branch reduction algorithm and algorithm based on dynamic maximum (DMDSE-Tree). These exact methods cannot satisfy the time and space requirements of engineering problems in large scale complex systems, and it is not necessary to find the optimal solutions in these systems. Therefore, heuristic algorithms have been proposed to approximate the optimal minimal hitting sets using algorithms like discrete particle swarm algorithm, binary particle swarm optimization, and low-cost heuristic algorithm STACCATO based on heuristic functions.

Atashpaz-Gargari et al. proposed a population-based colonial competitive algorithm (CCA) by simulating the process of colonial competition in human society evolvement. It shows better convergence rate when compared to genetic algorithm and particle swarm optimization and has received more and more applications in recent years. For example, Forouharfard et al. used the colonial competitive algorithm to solve the logistic planning problem in transshipment warehouses. Kaveh et al. applied the colonial competitive algorithm to the structural optimal design problem. Nazari et al. employed the colonial competitive algorithm for the product outsourcing problems in mix-model production. Sarayloo et al. applied the colonial competitive algorithm in solving the dynamic unitised manufacturing problem. In addition, numerous attempts have been made to improve the performance of colonial competitive algorithm and apply it to other combinatorial optimization problems, including mixed-model assembly line balancing problem and the data clustering problem. In this paper, we propose an improved colonial competitive algorithm to solve the minimal hitting set problem and also validate its performance using the enterprise equipment selection problem.

2 Improved colonial competitive algorithm for the minimal hitting set problem

In this paper, we propose an improved colonial competitive algorithm by introducing the concept of independent countries in addition to the existing two types of countries, namely, empires and colonies, in classical CCA. They are defined as follows:

- empire: empire countries strive to assimilate more colonies in the computational process of the algorithm.
- colony: colony countries aim to learn from their corresponding empires in order to become either independent countries or new empire countries.
- independent country: independent countries try to learn from and turn into empire countries.

The main differences between the improved CCA from the classical one exist in the country initialization, assimilation and update stages due to the introduction of independent countries in the algorithm workflow.

2.1 Empire initialization

The improved CCA requires three parameters, namely, N_{imp}, N_{col} and N_{ind}, to indicate the number of empires, colonies and independent countries in the initial population. Therefore, the total number of countries in the initialization step is $N_{pop} = N_{imp} + N_{col} + N_{ind}$. The algorithm first randomly creates N_{pop} solutions (countries) and selects the best N_{imp} countries as empires, the rest countries will be marked as colonies and independent countries based on the parameters N_{col} and N_{ind}. Note that independent countries are not influenced by any empire and therefore not part of any empires. Figure ? depicts the country classification in the initialization stage.

2.2 Solution encoding

Solution encoding scheme is the key step in applying colonial competitive algorithm to optimization problems and will greatly affect its performance thereafter. Binary encoding theme is the most widely used method since it is easy to implement.

For a set group U, an assimilation step is first conducted to remove all the sets that encompass any other sets, which can greatly reduce the number of hitting sets. Only the encompassed sets, denoted by U_i, are left after the assimilation process. Let n indicate the number of sets available in the set group after the assimilation process. Let R represent the union of all sets in the set group, that is, $R = \cup_{i=1,2,\cdots,n} U_i$, $|R| = N$. It is clear that the minimal hitting set is a proper subset of R and every element in R either exists or does not exist in the minimal hitting set. Therefore, a binary value can be used to indicate whether an element in R shows up in the minimal hitting set. In this way, the minimal hitting set can be represented by an array of length N and each element in the array must be a binary value of either 0 or 1.

In order to create good approximations of the minimal hitting set in the initial population, reduce the number of algorithm iterations, and decrease the minimizing calculations, the average number of array elements with value of 1 is set to be close to the number of elements in the minimal hitting set. Let X

denote the average number of elements in the minimal hitting set and β be the probability that an element takes the value of 1, then,

$$\beta = X/N \tag{1}$$

Since the value of X is unknown in general, the value of β must be approximated, which is empirically set to $0.1 \sim 0.5$. Note that the value of β increases when n increases.

Let Y be a chromosome and the gene at the ith ($i < N$) position be x_i, then $Y = \{x_1, x_2, \cdots, x_N\}$. The following method is used to create the initial solution based on the binary encoding scheme: randomly create a number $\xi_i \in (0, 1)$, let $x_i = 1$ if $\xi_i < \beta$; let $x_i = 0$ if $\xi_i \geq \beta$.

2.3 Fitness function

The definition of fitness function plays a key role in deciding an algorithm's search efficiency.

2.4 Minimizing operators

2.5 Competition

2.6 Assimilation

2.7 Update

2.8 Empire removal

References