ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

BCO for Multiprocessor Scheduling

K. Sireesha
Student, Dept. of Information Technology, LBRCE, JNTUK University, Andhra Pradesh, India

Abstract
In the multiprocessor scheduling problem, a given list of tasks has to be scheduled on identical parallel processors. Each task in the list is defined by a release date, a due date and a processing time. The intent is to minimize the number of processors utilized while respecting the constraints imposed by release dates and due dates. This aim is clearly linked with minimizing the cost of hardware required for implementing a specific application. The scheduling problem is a key factor for distributed systems to gain better performance. In this paper, we have proposed a metaheuristic approach BCO for this problem. On the standard benchmark instances for the problem, performance of our approach is comparable to other state-of-the-art approaches.

Key words
Multiprocessor scheduling problem, release date, due date, processing time, BCO

I. Introduction

A list of n tasks is given and these tasks have to be scheduled on identical parallel processors (each processor has the same computing ability). Each task i has a release date ri, a due date di, and a processing time pi. Let si be the start time of task i. A task i can start only after its release date, i.e., ri and has to be finished before its due date di. The objective of multiprocessor scheduling problem considered in this paper is to schedule the tasks so as to minimize the number of processors used in such a way that the constraints imposed by release dates and due dates should not be violated. This objective is directly linked with minimizing the cost of the hardware needed for implementing a specific application [5, 8], where tasks must be executed on the cheapest possible architecture.In order to interpret scheduling problem, various methods have been proposed. The proposed methods can be normally classified into three categories: Graph-theory-based approaches, mathematical models-based methods and heuristic techniques. Task scheduling in distributed systems is called as NP-hard. Therefore applying heuristic techniques can solve this problem more efficiently. Three most well-known heuristics are the iterative improvement algorithms, the probabilistic optimization algorithms, and the constructive heuristics. In the probabilistic optimization group, GA-based methods are noticeable. The prime distinction among them is chromosomal

representation needed for a schedule. However, these approaches scan the entire solution space without considering the techniques that can diminish the complexity of the optimization. In other words, their main shortcoming is spending much time doing scheduling. This shortcoming of GA-based methods can be decreased by combing GA with another optimization technique. Hence, this paper introduces a new algorithm by using memetic algorithm to cope with this shortcoming. Bee Colony Optimization has been applied as local hunt in the proposed memetic algorithm. The results demonstrated that the proposed method comparable with the three existing methods in terms solution quality and running time. There are two categories for task scheduling: static and dynamic. In dynamic scheduling, schedules are generated during runtime. The details of the tasks like processing time, due date and release date are unknown. While in static scheduling, schedules are created before run time and do not change during the scheduling. In other way, tasks must be all known in advance. In other words, the details of the tasks like processing time, due date and release date are known before the scheduling process has begun. In this paper, the author concentrated on static scheduling.

The rest of this paper is organized as follows: Section 2 provides a brief introduction to Artificial Bee Colony (ABC) Algorithm. Section 3 a brief introduction to Artificial Weed Colony (ABC) Algorithm/Invasive Weed Optimization (IWO). Section 4 provides BCO approach to the multi-processor scheduling. Computational results are presented in Sect. 5, whereas Sect. 6 outlines some concluding remarks and directions for future research.

II. Abc

ABC is a population based meta-heuristic optimization algorithm [1-4, 6, 7, 10], which is inspired by the intelligent behavior of natural honey bee swarm. The natural bees are divided into three categories employed, onlooker and scout bees based on their work.

Employed bees: Collect the food from the food sources and bring it to the hive. After coming to the hive they share the information with the onlooker bees. Every employed bee associated with one food source.

Onlooker bees: They get information about food sources from employed bees. The food sources which are having high nectar amount attract the more bees. The employed bees share the information about food sources with the onlooker bees which are waiting in the hive. To share the information the employed bees perform dances in a common area in the hive, which is called dancing area. Onlooker bees observe the dances performed by employed bees before choosing a food source. The probability of food source to be chosen by the onlooker bees is proportional to the nectar amount of that food source. Therefore, the food sources which are having high nectar amount attract more bees than the other food sources which are having less nectar amount.

Scout bees: Whenever a food source completely exhausted, all the employed bees associated with it leave it, and become scouts. Scouts look for new food sources in the vicinity of the hive. Whenever a bee, whether it is scout or onlooker finds a food source then it becomes employed. So in a way exploration is done by the scouts and exploitation is done by the employed and onlookers .In a employed bee phase try to improve each and every solution called as deterministic approach. In this onlooker bee phase select any two solutions and improve the better solution among them, called as probabilistic approach. In the scout bee phase if a solution is not improved for certain number of iterations replace with a random solution.

III. Awc

A novel numerical stochastic optimization method is inspired from colonizing weeds. This optimization method has been introduced for optimizing antenna problems. This algorithm has been first used by Mehrabian and Lucas [9]. They have named this algorithm as Invasive Weed Optimization (IWO). Invasive Weed Optimization (IWO) is a population based algorithm which is based on trial and error method that simulates the colonizing behavior of weeds. The common phenomenon in agriculture is colonization of weeds. IWO is motivated by this phenomenon. A weed is considered as an undesirable plant in agriculture because of its robust and adaptive nature. Despite of being the algorithm is simple; it is effective in converging to the optimal solution because it is employing the basic properties of a weed colony such as seeding, growth and competition.

IV. Bco

Bee Colony Optimization (BCO) is a metaheuristic algorithm which is inspired by the intelligent foraging behavior of natural honey bee swarm for solving combinatorial optimization problems. In a honey bee colony forager bees explore the environment for food sources and if they find any food source, they share the information with the other bees which are waiting in the hive. To share the information, the bees perform dances in the common area called as dancing area. Once the forager bees come back to the hive they share the food sources information to the waiting bees by a special movement named waggle dance.

The probability of a food source to be chosen by the bees is proportional to the nectar amount of that food source. Therefore, the food sources which are having high nectar amount attract more bees than the other food sources which are having less nectar amount. When a food source is completely exhausted, all the employed bees associated with it leave it and look for a new food source.

The algorithm consists of two alternating phases: forward pass and backward pass.

To continue, it is the pseudo code of the BCO algorithm:

1. Initialization: Every bee is initialized with a solution
2. Every bee will go through the forward pass:
   a) Set i = 1; //counter value for constructive moves in the forward pass.
   b) The bee does evaluate all the possible constructive moves;
   c) According to the evaluation, choose the best among the possible moves
   d) i = i + 1; If i ≤ NC Go To step b.
3. All bees are back to the hive to share the information; // backward pass starts;
4. Sort the bees by their objective
5. The bee which has high objective function value will explore on its own, the other bees with less objective function value will become (bees with higher objective function value have greater chance to continue its own exploration);
6. For every follower, choose a new solution from recruiters
7. If the stopping condition is not met Go To step 2;
8. Output the best result.

The algorithm is initialized by assigning a randomly generated solution to each employed bee. The process of generation of an initial solution is described below.

This initial solution generation process tries to schedule the tasks as early as possible on one of the available processors. Only when it is not possible to schedule the tasks anywhere, a new processor is added to the solution and task is scheduled on it at the earliest possible time.

- A task is selected uniformly at random for allocation. The selected task is tried for insertion at the beginning of a processor.
- If it is not possible to insert at the beginning of a processor, then it is tried for insertion at the middle of a processor.
- If it is also not possible to insert the task in the middle of some processor, then it is tried for insertion at the end of a processor.
- When it is not possible to insert the task anywhere, then a new processor is added to the solution and the task is allocated to it. This process is repeated until all tasks are allocated.

V. Computational Results:

The MPS-BCO has been coded in C and were executed on a Linux based 3.0 GHz core2duo system with 2 GB of RAM. Tables 1 and 2 clearly show that the BCO approach returned better quality solutions in shorter time. Out of 100 instances it solves 84 instances optimally whereas ABC, AWC and HGGA respectively solve 90, 72 and 97 instances optimally.

Table 1: Comparison of MPS-BCO with MPS-ABC, MPS-AWC and HGGA in terms of solution quality

Instance size (n)	HGGA	ABC	AWC	BCO
	Instances solved to optimality			
20	20	20	20	20
40	20	20	16	20
60	20	19	15	17
80	19	18	12	16
100	18	13	9	11

Table 2: Average execution times of MPS-BCO with MPS-ABC, MPS-AWC and HGGA in seconds

Instance size (n)	HGGA	ABC	AWC	BCO
	Instances solved to optimality			
20	0.05	0.01	0.12	0.05
40	0.17	0.03	0.25	0.11
60	0.40	0.07	0.49	0.17
80	0.88	0.13	0.84	0.35
100	1.32	0.25	1.34	0.72

VI. Conclusions

In this paper, we have presented a metaheuristic approach Bee Colony Optimization, for a multiprocessor scheduling problem where the objective is to minimize the total number of processors used. This objective is directly linked with minimizing the cost of the hardware needed for implementing a specific application. We have compared our BCO approach with other metaheuristic approaches and, also steady-state grouping genetic algorithm. Our BCO approach is comparable with the other metaheuristic approaches and also grouping genetic algorithm in terms of solution quality as well as running time.

References

[1]. Akay, B., Karaboga, D.: A modified artificial bee colony algorithm for real-parameter optimization. Inf. Sci. 192, 120–142 (2012)

[2]. Basak, A., Maity, D., Das, S.: A differential invasive weed

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

optimization algorithm for improved global numerical optimization. Appl. Math. Comput. 219(12), 6645–6668 (2013)

[3]. Basturk, B., Karaboga, D.: An artificial bee colony (ABC) algorithm for numeric function optimization. In: Proceeding of the IEEE Swarm Intelligence Symposium, pp. 12–14. IEEE, Indianapolis (2006)

[4]. Bonabeau,E.,Dorigo,M.,Theraulaz,G.:Swarm intelligence: from natural to artificial systems. Oxford University Press, New York (1999)

[5]. Gajski, D.D., Dutt, N.D., Wu, A.C.-H., Lin, S.Y.-L.: High level synthesis: Introduction to chip and system design. Academic Publishers, Boston (1992)

[6]. Gao, W., Liu, S.: Improved artificial bee colony algorithm for global optimization. Inform. Process. Lett. 111(17), 871–882 (2011)

[7]. Gao, W., Liu, S., Huang, L.: A global best artificial bee colony algorithm for global optimization. J. Comput. Appl. Math. 236(11), 2741–2753 (2012)

[8]. Lee, J.-H., Hsu, Y.-C., Lin, Y.-L.: A new integer linear programming formulation for the scheduling problem in data path synthesis. In: Proceedings of the 1989 IEEE International Conference on Computer-Aided Design, pp.20–23. IEEE Computer Society Press, New York (1989)

[9]. Mehrabian, A., Lucas,C.: A novel numerical optimization algorithm inspired from weedcolonization. Ecol. Inform.1(4), 355–366 (2006)

[10] .Karaboga, D., Akay, B.: A modified artificial bee colony (ABC) algorithm for constrained optimization problems. Appl. Soft Comput. 11, 3021–3031 (2011)

Authors Profile

Koduru Sireesha pursuing M.Tech Degree in Software Engineering from Lakireddy Balireddy College of Engineering,Vijayawada,AP, INDIA. Her research Interests include Software Engineering, Image Processing and Biometrics.