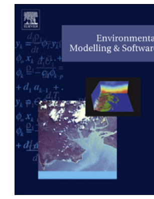
Environmental Modelling & Software 57 (2014) 152–164

Contents lists available at ScienceDirect

Environmental Modelling & Software

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



Parallel discrete differential dynamic programming for multireservoir operation

Chuntian Cheng [a,*], Sen Wang [a], Kwok-Wing Chau [b], Xinyu Wu [a]

[a] *Institute of Hydropower and Hydroinformatics, Dalian University of Technology, Dalian 116024, China*
[b] *Department of Civil & Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China*

ARTICLE INFO

Article history:
Received 2 May 2013
Received in revised form
25 January 2014
Accepted 28 February 2014
Available online 27 March 2014

Keywords:
Hydropower systems
Discrete differential dynamic programming
Parallel
Long-term operation
Optimization

ABSTRACT

The curse of dimensionality and computational time cost are a great challenge to operation of large-scale hydropower systems (LSHSs) in China because computer memory and computational time increase exponentially with increasing number of reservoirs. Discrete differential dynamic programming (DDDP) is one of the most classical algorithms for alleviating the dimensionality problem for operation of LSHSs. However, the computational time performed on DDDP still increases exponentially with increasing number of reservoirs. Therefore, a fine-grained parallel DDDP (PDDDP) algorithm, which is based on Fork/Join parallel framework in multi-core environment, is proposed to improve the computing efficiency for long-term operation of multireservoir hydropower systems. The proposed algorithm is tested using a huge cascaded hydropower system located on the Lancang River in China. The results demonstrate that the PDDDP algorithm enhances the computing efficiency significantly and takes full advantage of multi-core resources, showing its potential practicability and validity for operation of LSHSs in future.

1. Introduction

Multireservoir operations are one of complex and challenging tasks (Labadie, 2004) addressed by many researchers in past decades. Comprehensive methods and models that deal with a variety of problems about hydropower reservoirs operations are now available. Numerous classical algorithms, such as linear programming (Trezos, 1991; Reis et al., 2006; Azamathulla et al., 2008), nonlinear programming (Martin, 1983; Lund and Ferreira, 1996; Barros et al., 2003), network flow algorithm (Braga and Barbosa, 2001), dynamic programming (DP) or improved dynamic programming (Yakowitz, 1982; Kumar and Baliarsingh, 2003; Goor et al., 2011; Liu et al., 2011; Zhao et al., 2012a, b), and heuristic programming (Dariane and Momtahen, 2009; Malekmohammadi et al., 2009; Moeini and Afshar, 2013; Zhang et al., 2013a), have been applied to the multiple reservoirs operation problems with nonlinear and non-convex objective functions. The advantages and disadvantages of these classical approaches were specified in the reported literatures (Yeh, 1985; Labadie, 2004). The choice of methods is dependent on the operation tasks, available data, objectives and constraints. In the past decade, large-scale hydropower system optimization operations had become very prominent with the fast development in China and Brazil (Barros et al., 2001, 2003; Zambon et al., 2012; Cheng et al., 2012a, b, c). Especially in China, one of the countries that are rich in water resources, with the gross theoretical hydropower potential of 694 GW and technically exploitable installed capacity of 542 GW, there has been a rapid rate of development of hydropower systems. Now, the total installed capacity of hydropower has exceeded 200 GW, and the number of hydropower plants is more than 45,000. The number of large and medium-size hydropower plants operated by a central dispatching center is more than 100 and its total installed capacity has reached 50 GW. In the future 20 years, the number of hydropower plants operated by a regional dispatching center will be over 200 and its total installed capacity will surpass 140 GW. The challenges to the operation management of large-scale hydropower systems are tremendous in China. The greatest obstacle faced in the optimal operation of hydropower system remains the curse of dimensionality with increasing numbers of reservoirs, resulting in exponential increase of computer memory and computational time. In solving problems on large and complex hydropower systems, a general idea is to adopt methods which can reduce or alleviate dimensions. Progressive optimality algorithm (POA) (Howson and Sancho, 1975; Turgeon, 1981; Cheng et al., 2012c), discrete differential dynamic programming (DDDP) (Heidari et al., 1971; Chow

* Corresponding author.
E-mail addresses: ctcheng@dlut.edu.cn (C. Cheng), wangsen19861227@163.com (S. Wang), cekwchau@polyu.edu.hk (K.-W. Chau), wuxinyu@dlut.edu.cn (X. Wu).

http://dx.doi.org/10.1016/j.envsoft.2014.02.018
1364-8152/

et al., 1975; Tospornsampan et al., 2005) and dynamic programming successive approximation (DPSA) (Erkmen and Karatas, 1994; Opan, 2011) have been developed to overcome the incremental dimensionality of DP along with the distensible scale of the hydropower systems. POA is a computationally efficient method of reducing the dimensionality difficulties by mean of successive approximation using a general two-stage solution. DDDP realizes the reduction of the discretized number of state variables by iterative search in the constantly changing corridor. DPSA decomposes a multi-dimensional problem into a sequence of one-dimensional problems by optimizing over one state variable at a time (Labadie, 2004). However, these approaches also have some disadvantages for solution. POA and DDDP are sensitive to initial trajectories, and may converge to a local optimum in some situations. DPSA is difficult to be applied to solution of non-convex problems for which converge to even a local optimum is not guaranteed. In spite of these inherent weaknesses, the approaches mentioned above are still widely applied for the operation of large-scale hydropower systems because of their feasibility in the practical problems. Other approaches employ an integrated mix of formulations and solution methods to alleviate the curse of dimensionality. However, they can only be applied to optimal operation problems with relatively simple, small-scale constraints and objectives. For large and complex hydropower systems, it is very difficult to simplify both the objective function and constraints to either non-linear or linear optimization model format for direct mathematical solution. Therefore, the priority is given to the improved DPs such as POA, DDDP and DPSA for the large and complex hydropower systems.

Some simplified algorithms mentioned above have successfully reduced the dimensions or alleviate the curse of dimensionality, but the computational time cost still enormously increases with the number of hydropower plants for the optimization solutions. Especially, when the scale of hydropower system reaches a certain large degree, computational time cost is absolutely intolerable. Therefore, the computational efficiency is a great challenge to the operation of large-scale hydropower systems. Generally, there are two basic approaches to improve computational efficiency. One is to improve the classical algorithms or seek for new and good algorithms. The other is to use new computer techniques including hardware and software. This paper will address the latter and our focus is on the parallelization of dimensionality reduction methods, especially how to utilize the current popular multi-core resources.

Since the release of the first batch multi-core processors by IBM in 2001, Sun in 2004, and AMD in 2005, more and more cores are built into a single processor with the development of multi-core technology. Moreover, the increasing popularity of multi-core processors provides the necessary hardware basis for the implementation of computational tasks with fine-grained parallel mechanism. Hence, multi-core parallel computing technology (Zhu et al., 2013; Morell-Gimenez et al., 2013) is always the major research field of computer science. Parallel computation in multi-core environment means that the whole task is decomposed into numerous subtasks, and then subtasks are assigned to different cores in which subtasks can be executed independently, for speeding up the computational process. Nowadays, improving computational efficiency by parallelization is confirmed to be successful in many fields (da Silva and Finardi, 2003; Rouholahnejad et al., 2012; Jordi and Wang, 2012; Bryan, 2013; Zhao et al., 2013; Joseph and Guillaume, 2013). Therefore, the research of parallel optimization algorithms is very significant in order to solve the time-consuming problem for large-scale hydropower systems. For realizing the parallelization, a variety of parallel frameworks have been developed such as Fork/Join (Lea, 2000), Message Passing Interface (MPI) (Li et al., 2011; Wu et al., 2013) and Open Multi-Processing (OpenMP) (Innocenti et al., 2009; Neal et al., 2010). The choice of parallel frameworks is often dependent on the characteristics of the task, applicable conditions, compatibility of operating systems and the diversity of source languages between algorithms and frameworks. For the study in the paper, the last consideration is what we concerned. The Fork/Join framework is used to realize the parallelization of our Java programmed algorithm because the Fork/Join framework has been packaged as standard programs in Java version 7 and widely recognized as an appropriate platform to parallelize the algorithms of Java programmed.

Though Fork/Join framework is not the most effective technique with good parallel performance (Lea, 2000), it also has some conspicuous advantages for an attractive choice. Firstly, Fork/Join framework can divide a task into lots of subtasks recursively with divide-and-conquer strategy, which is suitable to be applied to the problems with heavy computational tasks such as optimization of hydropower system operation. Secondly, Fork/Join framework can make full use of multi-core resources which enables it to be applied widely in a popular personal computer with multiple cores. Thirdly, it possesses lightweight scheduling mechanics to be independent of any container. At last, Fork/Join framework is an open source program so that it is much easier to design parallel algorithms for the given problems.

In this paper, due to our focus on realization of the parallelization, the stochasticity of inflows has not been considered in the model for simplifying calculations, and DDDP is selected to test the computational efficiency of parallelization. A fine-grained parallel discrete differential dynamic programming (PDDDP) algorithm, which is based on Fork/Join parallel framework (Lea, 2000) in a multi-core environment, is proposed to improve the computational efficiency for long-term operation of multireservoir hydropower systems. The parallelization of DDDP is thoroughly analyzed. A huge cascaded hydropower system, located in the Lancang River which is one of the 13 largest hydropower bases in China, is used to test the feasibility and validity of the proposed algorithm. The case studies show that high parallel efficiency and rational optimization solution can be obtained by the use of PDDDP, and much less computer time is expended on the computational process than the original DDDP. Therefore, with the fast development of hydropower systems in China, it is expected that the parallelization of the optimization algorithms will be an effective approach to improve the computational efficiency for the operation of large-scale hydropower systems in future.

2. Problem formulation

2.1. Variable definition

Hydropower system operation is characterized by various objectives and constraints. The notations used in the paper are introduced as follows:

E	Total sum of energy produced by all reservoirs in time horizon [MWh]
T	Number of time steps
M	Number of reservoirs in hydropower system
p_m^t	Power generation of reservoir m for period t [MW]
Δ_t	Duration [h]
k_m	Generation efficiency of reservoir m
q_m^t	Average turbine discharge of reservoir m for period t [m^3/s]
H_m^t	Net head of reservoir m for period t [m]
$H_{m,s}^t$	Storage water level of reservoir m for period t [m]
$H_{m,b}^t$	Tail water level of reservoir m for period t [m]

(continued on next page)