

Performance Configuration Optimization of Hadoop System Based on Genetic Simulated Annealing Algorithm

Wanying Bao, Xiaoling Luo*

College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot Inner Mongolia
Email: *baowy1995@163.com

Received: Apr. 21st, 2020; accepted: May 6th, 2020; published: May 13th, 2020

Abstract

In order to improve the performance of Hadoop on open-source cloud computing platform, a performance configuration optimization method based on genetic simulated annealing algorithm is proposed. Based on genetic algorithm, the configuration scheme is selected, crossed and mutated as chromosome. Combined with the principle of simulated annealing, the survival rate of new chromosome and the number of iterations of the whole algorithm are controlled to find the optimal scheme of system configuration. According to the genetic simulated annealing algorithm, the overall performance is better, and the optimization speed is faster in the long-term optimization. It can be used to solve the problem of finding the approximate optimal allocation scheme of the system through random search in the global space. Experimental results show that this method can effectively improve the efficiency of finding the optimal configuration. The proposed configuration method improves the operation speed, makes full use of resources and increases the throughput of the system.

Keywords

Genetic Simulated Annealing Algorithm, Hadoop, System Performance, Configuration Optimization

基于遗传模拟退火算法的Hadoop系统性能配置优化

包婉莹, 罗小玲*

*通讯作者。

内蒙古农业大学计算机与信息工程学院, 内蒙古 呼和浩特
Email: *baowy1995@163.com

收稿日期: 2020年4月21日; 录用日期: 2020年5月6日; 发布日期: 2020年5月13日

摘要

为提高Hadoop在开源云计算平台上的性能, 提出了基于遗传模拟退火算法的Hadoop系统性能配置优化方法。基于遗传算法, 将配置方案作为染色体进行选择, 交叉和变异。结合模拟退火原理, 控制新染色体的存活率和整个算法的迭代次数, 找到系统配置的最优方案。根据遗传模拟退火算法得到的整体性能较好, 在长期优化中优化速度更快, 可以用来解决在全局空间内通过随机搜索找出系统的近似最优分配方案的问题。实验结果表明, 该方法可以有效提高寻找最优配置的效率。提出的配置方法提高了操作运行速度, 充分利用了资源, 增加了系统的吞吐量。

关键词

遗传模拟退火算法, Hadoop, 系统性能, 配置优化

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 前言

随着网络的迅速发展, 视频和照片等在线观看和共享服务的数量逐步增加, 网站需求分析和日志处理的数据也在增加。为实现日益增加的海量数据的处理, 上个世纪七八十年代, 人们开始研究并使用并行计算来处理一些复杂问题。到了九十年代中期开始有了对网格计算的应用。随着这些高性能计算方式的不断演化, 云计算发展起来。类似谷歌云计算平台, Windows Azure 云计算架构和亚马逊云计算平台等许多云计算平台不断涌现。Hadoop 作为开源的云计算平台, 实现了 Google 云计算的功能, 被研究者们广泛使用[1]。Hadoop 系统是一个可以运行在一台机器或者几千台普通的商用服务器作为节点组成的集群上, 可以将海量数据集进行分布式运算的开源框架。目前, 它已经应用于 Yahoo, FaceBook 等企业, 许多研究人员正在研究 Hadoop 平台, 为提高 Hadoop 的性能, 使它可以更好地提供高质量的服务。

2. 概述

2.1. 云计算技术

云计算通过集成、管理和分发分布在互联网上的所有计算资源来同时向用户提供服务。云计算具有规模大、泛在接入、弹性服务、按需服务、便宜等主要特征。云计算的成功与虚拟化技术、分布式文件系统、并行计算模型和非关系数据库等技术的支持密切相关[2]。分布式开源云计算平台 Hadoop 是 Apache 的一个项目, 主要用于存储和处理海量数据。Hadoop 提供给组件分布式系统的工具包括数据存储、数据分析等[3]。Hadoop 可以将大量程序和数据分发给数十万台普通计算机, 每台计算机作为分布式集群的节点运行, 所有节点构成一个大型数据中心。Hadoop 的主要思想是使用函数式编程语言, 通过集群分解数据和任务, 提高处理速度, 备份和容错技术用来保证操作的安全性。

2.2. Hadoop 体系结构

Hadoop 可以将大量程序和数据分发给数十万台普通计算机, 每台计算机作为分布式集群的节点运行。Hadoop 的主要思想是从函数式语言(比如 Lisp)中的内置函数 map 和 reduce 中得到, 并将海量且结构复杂数据和任务分发到集群中每一个机器上, 加快数据处理速度, 同时具有备份和容错机制。Hadoop 平台的架构如图 1 所示。

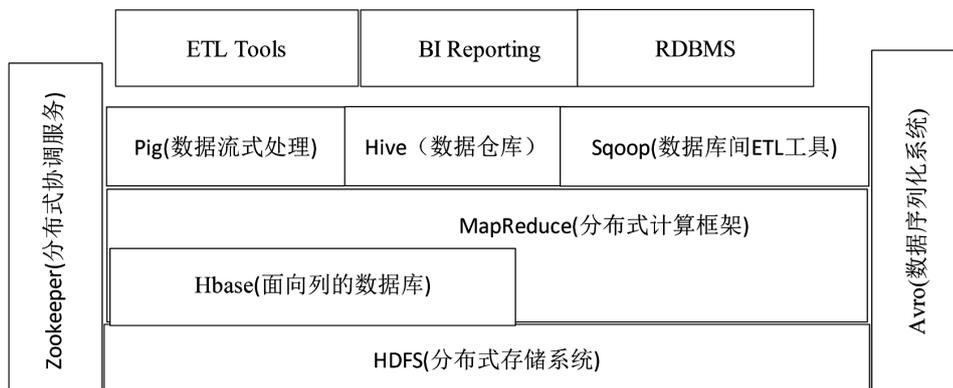


Figure 1. Hadoop platform architecture

图 1. Hadoop 平台架构图

Hadoop 的核心是 HDFS 分布式文件存储和 MapReduce 计算处理框架。这两个主要组件的使用使得 Hadoop 具有可拓展、成本低、安全性高、高效率等优点。此外, Hadoop 还包括 HBase、Pig、Hive、Sqoop、Zookeeper 和 Avro。这些工具的配合使 Hadoop 能够与当前的数据库系统进行数据交换, 成为在海量数据计算领域具有文件系统、计算模型、数据库系统和管理模块等综合功能的平台。

3. 架构与方法

3.1. Hadoop 性能优化

在 Hadoop 系统中, 系统的所有管理都是通过参数配置完成的。使用 Hadoop 时, 用户必须首先配置系统参数。作业的类型不同则对系统资源的需求也有所差别, 为提高运行效率, 吞吐量和系统的整体性能, 每种需求都需要进行单独的参数配置。如果仅仅依赖系统的静态默认配置则无法满足所有程序的资源最优。因此, 通过研究配置优化方法来改进系统的整体性能具有十分积极的意义。

首先分析影响 Hadoop 系统性能的主要配置属性及其之间的约束关系, 提出了基于遗传模拟退火算法的性能优化方法。其次, 考虑到系统中应用程序的复杂性, 提出了一个自适应系统配置优化策略。为适应系统中操作类型的复杂性, 该策略使用“实时优化”配置模型和“经验借鉴”配置模型来配置系统参数。

3.2. Hadoop 配置属性分析及其基准程序

Hadoop 系统架构主要由 HDFS 文件系统和 MapReduce 计算模型以及默认配置文件组成。在 Hadoop 集群中, 需要配置的文件主要包括 3 个, 分别 core-default.xml, hdfs-default.xml 和 mapred-default.xml。这些属性的配置支撑着整个系统的运作。在不影响系统的默认配置的前提下, 可以对相应的站点文件中这三部分的默认属性值进行修改。表 1 和表 2 列出了 Hadoop 系统的基本配置属性和影响系统性能的主要配置属性。

Table 1. Hadoop basic configuration properties**表 1.** Hadoop 基本配置属性

配置文件	配置属性	说明
core-site.xml	fs.default.name	Address of hdfs
	hadoop.tmp.dir	Storage folder of hdfs
hdfs-site.xml	dfs.replication	Backups of the data in hdfs
	dfs.http.address	http accessing address of hdfs
mapred-default.xml	mapred.job.tracker	Address of jobtracker

Table 2. Main configuration attributes affecting Hadoop performance**表 2.** 影响 Hadoop 性能的主要配置属性

配置文件	配置属性	说明
core-site.xml	io.file.buffer.size	使用缓冲区的排序文件的大小
	io.compression.codecs	数据压缩编码
hdfs-site.xml	dfs.block.size	文件数据块的大小
	dfs.namenode.handler.count namenode	远程调用的线程数
mapred-default.xml	dfs.heartbeat.interval	每秒心跳次数
	mapred.reduce.tasks	每项工作的 reduce 个数
	mapred.compress.map.output	map 输出数据是否压缩
	io.sort.factor	文件排序时合并的流数
	io.sort.mb	文件排序时缓冲区的大小

从配置 Hadoop 参数对进行 Hadoop 性能调优的角度而言, 首先要考虑的是集群自身的特点, 集群特点不同所以会有不同的参数设置。其次, 集群中的参数与参数之间也具有比较复杂的制约关系, 配置时要注意这些关系不被破坏, 保持整体的相对平衡。此外, 应用任务要在合适的执行环境下运行, 这对环境也提出了不同的要求, 因此需要针对任务的需要相应的调整系统配置实现任务的高效运行。

作为大数据分析平台的代表, MapReduce 拥有巨大的参数空间, Hadoop 的配置属性有超过 180 个, 所有的配置属性以键值对的形式存储在 XML 格式化文件中。其中有 70 多个参数影响作业性能, 直接造成 MapReduce 作业参数的调整艰巨而费时[4], 表 3 是影响 Hadoop 性能的部分配置属性和默认值的示例。以下面两个属性为例分析属性配置对资源利用的影响。

Table 3. Some properties and default values that affect Hadoop performance**表 3.** 影响 Hadoop 性能的部分属性和默认值

配置属性	默认值
dfs.datanode.handler.count	3
dfs.replication.min	1
dfs.namenode.handler.count	10
io.sort.factor	10
io.sort.mb	100
io.sort.record.percent	0.05
io.sort.spill.percent	0.80
mapred.job.tracker.handler.count	10
mapred.map.tasks	2
mapred.reduce.tasks	1
mapred.reduce.parallel.copies	5

每个 Map/Reduce 任务都作为一个单独的进程运行, 并且多个任务是并行处理的。如果并行运行进程太多, 则会导致资源争用并降低系统的整体性能。如果这两个值设置得太高, 将生成大量的磁盘同步读写操作并且争用磁盘资源。恰当的设置每个资源的 map/reduce 任务, 平衡配置属性之间的相互约束才能够使资源可以充分利用, 系统的性能达到最佳。

目前它主要依靠经验和暴力搜索来寻找最佳配置, 这需要很长时间并且效果不显著。一旦集群的结构或应用变化很大, 现有配置将不再存在申请, 需要重新积累经验。因此, 考虑一种新的配置方法来提高系统的性能是十分必要的。

基准测试程序通过测试集群按照预期运行任务来检测集群的性能, 测试的结果用于调整集群的性能达到最佳状态。不同类型的工作对各种资源都有不同的需求, 根据实际操作选择基准程序的类型, 从而调整各种配置参数并使其性能系统实现最佳。表 4 列出了一些 Hadoop 提供的基准测试程序。

Table 4. Hadoop partial basic test program

表 4. Hadoopde 部分基础测试程序

基准程序	测试性能
DFSCIOTest	分布式 I/O
dfsthroughput	HDFS 的吞吐量
testfilesystem	文件系统的读写测试
testmapredsort	Map Reduce 整体
Gridmix	集群加载功能
MRBench	复杂查询的关系数据处理和执行能力
NNBench	名称节点负载

前四个测试程序在 hadoop 安装目录下的 `hadoop -* -test.jar` 中。这些程序可以很容易地以最低成本进行测试。Gridmix 就在 Hadoop 安装目录下的 `src/benchmarks /` 目录中。与表 4 前四个测试程序相比, Gridmix 模拟了以 `webdataScan`, `javaSort` 和 `streamSort` 为代表的实际集群的工作负载。MRBench 面向业务数据的处理和大量数据的修改, 主要处理关系数据并执行复杂的查询。NNBench 用于测 namenode 的负载, 英特尔公司还提出了一套 Hadoop 基准测试套件 HiBench。表 5 是对测试用例的分析和 HiBench 中包含的资源特征。

Table 5. HiBench test case and its resource characteristics

表 5. HiBench 测试用例和它的资源特征

应用	基准程序	资源特征
Micro benchmarks	Sort	Intensive I/O, middle CPU
	Word count	Intensive CPU, networkand light I/O
	Tera sort	Map and Shufffle stage is intensive CPU
Web search	Nutch indexing	Map stage is intensive CPU Reduce stage is intensive I/O
	Page rank	Intensive CPU
Machine learning	Bayesian	Intensive I/O
	Classification	The first job is intensive CPU
	K-means clustering	Center point intensive CPU Clustering is intensive I/O
HDFS	Enhanced DFSIO	Intensive I/O

3.3. 实验方法

遗传模拟退火算法(GA)是一种将遗传算法与模拟火算法相结合的智能算法。遗传算法是一类借鉴生物进化规律演化而来的随机搜索方法, 常用于寻优, 但易陷入局部极值[5], 模拟退火算法[6]是一种通过模拟热力学中固体退火原理的随机寻优算法, 不易陷入局部最优。因此, 提出了结合遗传算法和模拟退火算法的遗传模拟退火算法[9], 遗传模拟退火算法是一种基于组合思想而产生的混合优化算法[7]以遗传算法为框架, 在执行遗传操作的过程中加入退火操作[8]。结合这两种算法, 不仅可以提高本地搜索能力, 也可以全球化优化, 找到理想的近似最优解。使算法能更好地跳出局部极值, 搜索到全局最优解[9]。

根据遗传模拟退火算法得到的整体性能较好, 在长期优化中优化速度会更快特点, 可以用来解决在全局空间内通过随机搜索找出系统的近似最优分配方案的这一类问题。以下将描述基于 Hadoop 配置的优化遗传模拟退火算法。

详细介绍主要操作: 生成初始种群, 染色体使用浮动编码方法, 每个基因位置的值范围(Min, Max), 随机生成的基因值是 $Min + (Max - Min) * Random()$;

选择: 轮盘赌选择方法用于生成随机数 $\gamma \in (0, 1)$ 。 f_i 是一个个体, 个体相对适应度是 $P_i = \frac{f_i}{\sum_{i=1}^0 f_i}$, 如

果 $\sum_{k=0}^{i-1} P_k < \gamma < \sum_{k=0}^i P_k$, 则 i 被选中复制。

交叉: 根据实数编码规则, 从种群中选择两个个体以一定概率交叉[10], 将染色体随机分成 22 组, 用概率 P_c 杂交, 染色体 A 通过算数杂交方法相交新的染色体 B, 染色体随机分成组, 每组包含 2 个染色体, 通过概率 P_c 执行交叉操作, 染色体组(a, b)通过算数杂交方法生成新的染色体组(a, b)。

$$a' = (1 - \alpha)a + \beta * b \quad b' = (1 - \beta)b + \alpha * a \quad (1)$$

在公式中, $0 < \alpha, \beta < 1, \gamma \leq 1$ 。如果 $a'(b') < Min$, 那么 $a'(b') = Min$ 。如果 $a'(b') > Max$, $a'(b') = Max$ 变异: 基因突变以 P_m 的概率进行, 基因 c 通过非均变异法形成 c' :

$$c' = \begin{cases} c + k * (Max - c) * \gamma\theta > 0.5 \\ c' = c + k * (c - Min) * \gamma\theta \leq 0.5 \end{cases}$$

在公式中, $k \in (0, 1]$, γ 和 θ 是(0,1)间的随机数。

4. 实验和结果分析

本实验主要使用 Hadoop-0.20.2 平台, 选择 17 个节点。每个节点配置如下: 2.2 GHz \times 2 CPU, 512 MB 内存, 40 GB 硬盘磁盘, RedHatLinuxAS4.0 操作系统。选择 WordCount 作为基准程序, 并且输入工作的大小是 0.6 G; 实验使用的是 Hadoop 默认的输入方法 TextInputFormat, map 的输入类型为 $\langle LongWritable, Text \rangle$, 输出的 key 值类型是 Text, 输出的 value 值类型是 IntWritable。map 方法对输入的行以空格为单位进行切分, 然后使用 OutputCollect 收集输出。

Reduce 类以 map 的输出作为输入, 因此 Reduce 的输入类型是 $\langle Text, IntWritable \rangle$, 它的输出类型是 $\langle Text, IntWritable \rangle$ 。Reduce 类也要实现 reduce 方法, 在本实验中, reduce 函数将输入的 key 值作为输出的 key 值, 然后将获得多个 value 值加起来, 作为输出的值。

GeneName, GeneRange 和 GeneType 三个文件都是在进行配置属性值的的规范化和染色体基因值的反规范化时使用。分别用来保存配置属性, 属性值范围和属性值类型, 通过这三个文件, 任何类型的属性值都规范化为浮点类型。算法运行时, 种群信息记录在日志文件中。

图 2 是在算法的每次迭代期间的最短运行时间, 最长运行时间以及作业的总体平均运行时间。从中

可以看出当迭代次数为 60 次时, 所得到的解决方案基本上接近于近似值最佳解决方案, 根据最长运行时间和最短运行时间曲线, 算法可以保证整体朝着最佳解决方案发展。在系统的默认配置下的操作时间为 92,476 ms。算法的最小运行时间是 71,744 毫秒, 运行效率为 26.4%。

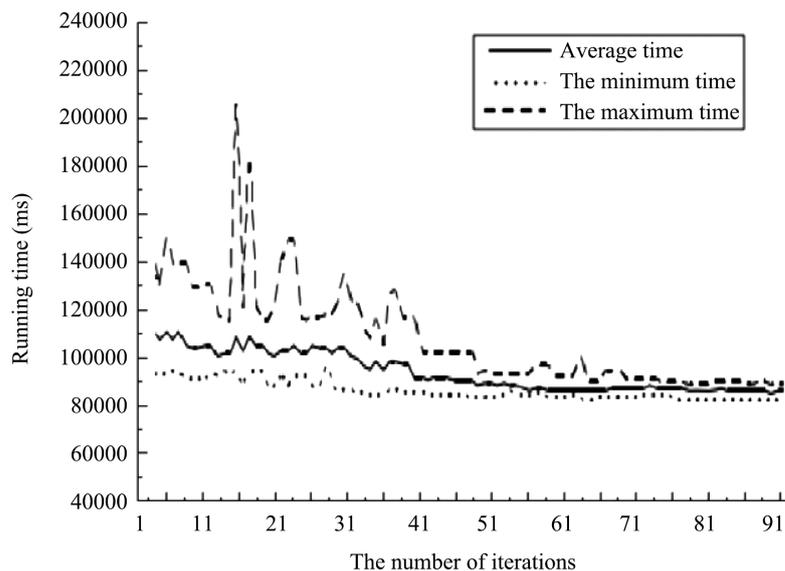


Figure 2. Algorithm optimization when the task size is 1.2 g
图 2. 当任务大小为 1.2 G 时的算法优化

为了验证最优方案具有一定的稳定性, 同样适合类似的工作。选择用相同的规格对 8 个不同的任务进行实验。在最佳配置方案下, 我们运行任务并记录运行时间。如表 6 所示, 平均提高效率为 24.1%, 表示配置方案可以真正提高运营效率。同时, 通过这个算法得到的最优配置的操作时间节省率几乎是相同的, 验证了配置方案是适用于类似工作且稳定的。

Table 6. Compare the best configuration and default configuration under different working conditions
表 6. 比较不同工作状态下的最佳配置和默认值配置

运行时间(ms)	优化配置	默认配置	提高效率(%)
Operation 1	73564	108107	31.9
Operation 2	83083	103638	19.8
Operation 3	76713	104069	26.3
Operation 4	75360	97941	23.1
Operation 5	80454	104800	23.2
Operation 6	80004	107720	25.7
Operation 7	76296	97320	21.6
Operation 8	74454	93383	20.3
Average	77491	102122.3	24.1

分析用遗传模拟退火算法找到近似 Hadoop 系统的最优配置的这一实验, 实验结果表明提出的配置提高了操作运行速度, 充分利用了资源, 增加了系统的吞吐量, 总体而言集群的性能近似最佳。该证明方法稳定适用。

5. 讨论

5.1. 配置优化方法的比较

表 7 列出了基于遗传模拟退火算法的配置优化方法和基于混沌粒子群优化的配置优化方法两种算法的主要区别。在优化过程中, 种群所有的染色体都在逐渐接近最优解, 并且收敛速度越来越慢。整体表现都在变好, 而且经过一定的经验后搜索速度更快。在混沌粒子群优化粒子使用 `gbest` 和 `pbest` 不断移动来搜索全局最优解。因此, 收敛速度更快。在 Hadoop 系统, 许多工作都有类似的资源要求。他们对系统的配置要求也大致相似。因此, 新的应用被添加到种群中, 我们可以继续使用基于遗传模拟退火算法对相似的种群搜索, 可以加速搜索新应用最佳配置的过程。总的来说, 这两种方法各有利弊。因此, 不同的优化方法用于不同的情况。根据系统的要求, 系统会通过适应的配置优化。

Table 7. Comparison between genetic simulated annealing algorithm and chaos particle swarm optimization algorithm

表 7. 遗传模拟退火算法与混沌粒子群算法的比较

对比因素	遗传模拟退火算法	混沌粒子群算法
实现	复杂	简单
操作方法	选择, 交叉, 变异	粒子飞行
优化程度	整体优化	同时记录整体优化和个体最优化
收敛率	慢	快
是否容易陷入本地优化	是	否

5.2. 自适应配置

通过以上分析, 可以看出配置优化方法是针对特定的应用来评估性能优化。实验结果表明该算法可以在很大程度上提高 Hadoop 系统的性能整体效果。

6. 结论

Web2.0 时代是一场互联网产业的升级换代。每天面临着网页浏览, 图像上传和下载以及视频浏览等大量的数据要处理, 这就需要打破传统观来应对信息爆炸的形势。云计算的出现已经改变了传统数据处理, 文件存储和数据访问的方式。在开源云计算平台上对 Hadoop 的性能提出了一种优化方法。本文所做的工作包括: 介绍云计算的概念和特征及 Hadoop 系统并且总结支持云计算的相关技术; 考虑系统配置对 Hadoop 的整体性能的影响, 针对具体应用提出遗传模拟退火算法和混沌粒子群优化算法两种系统配置优化方法, 基于以上提出了适用于不同应用场合的自适应配置优化模型。

基金项目

内蒙古自治区自然科学基金(2019MS06015)。

参考文献

- [1] 吴浩宇. 基于 Hadoop 的同源性搜索 GO 功能注释平台的研究[D]: [硕士学位论文]. 南京: 南京农业大学, 2013.
- [2] 柳香. 云计算环境下的计算模型性能优化研究[D]: [硕士学位论文]. 石家庄: 河北师范大学, 2012.
- [3] 杨润芝, 肖卫青, 胡开喜, 等. 云计算平台上实现 30 年气候资料整编的方法[J]. 计算机技术与自动化, 2013, 32(3): 137-140.
- [4] 童颖. 基于机器学习的 hadoop 参数调优方法[D]: [硕士学位论文]. 武汉: 华中科技大学, 2016.

-
- [5] KEI-Shorbagy, M.A., Ayoub, A.Y., El-Desoky, I.M. and Mousa, A.A. (2018) Anovel Genetic Algorithm Based K-Means Algorithm for Cluster Analysis. In: Hassanien, A., Tolba, M., Elhoseny, M. and Mostafa, M., Eds., *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*. *Advances in Intelligent Systems and Computing*, Springer, Cham, 92-101. https://doi.org/10.1007/978-3-319-74690-6_10
- [6] Gao, C., Gao, Y. and Lv, S. (2016) Improved Simulated Annealing Algorithm for Flexible Job Shop Scheduling Problems. *2016 Control and Decision Conference*, 2191-2196.
- [7] Kang, Z. and Qu, Z. (2017) Application of BP Neural Network Optimized by Genetic Simulated Annealing Algorithm to Prediction of Air Quality Index in Lanzhou. *2017 2nd IEEE International Conference on Computational Intelligence and Applications*, Beijing, 8-11 September 2017, 155-160. <https://doi.org/10.1109/CIAPP.2017.8167199>
- [8] 郭彩杏, 郭晓金, 柏林江. 改进遗传模拟退火算法优化 BP 算法研究[J]. 小型微型计算机系统, 2019, 10(10): 2063-2067.
- [9] 杨从锐, 钱谦, 王锋, 等. 改进的自适应遗传算法在函数优化中的应用[J]. 计算机应用研究, 2018, 35(4): 1042-1045.
- [10] 陈闯, Ryad Chellali, 邢尹. 改进遗传算法优化 BP 神经网络的语音情感识别[J]. 计算机应用研究, 2019, 36(2): 344-346+361.