

引用格式:林涛,郝章肖,冯竞凯. 基于改进乌鸦搜索算法的云计算任务调度研究[J]. 微电子学与计算机, 2020, 37(2): 20-24.
[LIN T, HAO Z X, FENG J K. The optimal entropy threshold image segmentation of the new fruit fly optimization algorithm [J]. Microelectronics & Computer, 2020, 37(2): 20-24.]

基于改进乌鸦搜索算法的云计算任务调度研究

林 涛, 郝章肖, 冯竞凯

(河北工业大学 人工智能与数据科学学院, 天津 300401)

摘 要: 提出了一种以云任务的完成时间和成本为优化目标的改进乌鸦搜索算法(IMCSA)的任务调度方法. 首先采用反向学习初始化种群, 在选择被跟踪乌鸦时根据记忆的适应度值择优选取, 避免了盲目性; 其次在位置更新过程中, 将乌鸦的位置与其反向学习得到的位置进行交叉, 择优选取, 能够有效提高收敛速度. 最后通过 CloudSim 平台与粒子群算法、遗传算法、Min_Min 算法和 CSA 进行对比, 结果表明 IMCSA 在不同实验下, 在任务完成时间和成本取得的效果均优于对比算法.

关键词: 云计算; 乌鸦搜索算法; 任务调度

中图分类号: TP301

文献标识码: A

文章编号: 1000—7180(2020)02—0020—05

Research on cloud computing task scheduling based on improved crow search algorithm

LIN Tao, HAO Zhang-xiao, FENG Jing-kai

(School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China)

Abstract: A task scheduling method based on the improved crow search algorithm (IMCSA) is proposed, which take cloud task completion time and cost as the optimization goal. Firstly, the initial population was generated by opposition-based learning, and the preferred value was selected according to the fitness value of memory when selecting the tracked crow. That can avoid blindness. Secondly, to effectively improve the convergence speed, the position of crow was crossed with the position obtained by opposition-based learning and chose the better position during the position update process. Finally, compared with particle swarm optimization, genetic algorithm, Min_Min algorithm and CSA under the CloudSim. The experimental show that the algorithm is superior to the contrast algorithm at the task completion time and cost under different experiments.

Key words: cloud computing; crow search algorithm; task scheduling

1 引言

近年来,关于云任务调度问题,有很多研究者做了研究. 齐平等^[1]针对云计算环境下并行任务因受资源影响而无法完成的问题,解决了资源失效问题,将改进的粒子群算法与资源可靠性评估模型相结合的调度算法,提高了云服务的可靠性,但是并未

考虑任务的完成时间和与成本. 文献[2]提出了基于正交田口的猫群优化算法(OTB-CSO),该算法将田口正交方法结合到 CSO 中的跟踪模式中,使得任务的完成时间和负载均衡性得到了很好提升,但是缺乏对能耗和成本等目标的优化. 文献[3]中提出了一种混合蚁群粒子群算法,用来提高资源调度的效率,缩短了任务调度过程中时间的消耗,但是没有考虑

到任务的执行成本,且对比算法较少.文献[4]针对任务调度过程出现的资源负载不均的情况,改进了蚁群算法中信息素更新规则,提出了基于负载平衡的蚁群优化算法(LBACO),避免了资源浪费和负载不均问题,但是没有考虑完成任务的成本.

Alireza 等人^[5]于2016年提出乌鸦搜索算法(Crow Search Algorithm, CSA),乌鸦搜索算法是仿照乌鸦觅食过程中乌鸦群体之间窃取和反窃取食物的行为.相比较其他群体智能算法而言,CSA 相对简单,仅有两个可调参数(飞行长度和感知概率),并且在不同的工程领域,CSA 比 GA、PSO、HS 等具有更好的效果. Gehad 等人^[6]将混沌映射应用于算法中,提出了混沌乌鸦搜索算法(CCSA)来克服收敛速度较低,并且容易限于局部最优的问题,并将 CCSA 应用到寻找最佳特征子集的问题上,并取得很好效果. Farid 等人^[7]在解决非凸经济负荷调度(ELD)问题时,通过增加优先级进行选择乌鸦和自适飞行长度(fl)来更好的解决 ELD 问题. Primitivo 等人^[8]提出了一种动态感知概率和依据 Levy 飞行实现随机扰动来在解决能源优化问题. 刘雪静等人^[9]提出了基于 Levy 飞行策略的混沌乌鸦搜索算法(LDECSA)结合二进制编码,用于求解{0-1}背包问题.

本文将改进的乌鸦搜索算法与云计算任务调度模型相结合用于解决云任务调度问题,以寻找最优的调度方案,使得任务的完成时间短,计算成本低.

2 云计算任务调度模型

2.1 任务调度描述

云计算任务调度模型^[10]主要分为4部分:任务、任务管理器、任务调度器和云环境.云计算任务调度一般从以下方面进行研究:一是任务-虚拟机的关系映射,即当用户提交任务到数据中心时,将任务更加合理地分配给虚拟机;二是虚拟机-物理机的关系映射,即选择合适的主机来执行虚拟机接收的任务.本文主要研究的是任务-虚拟机的映射.

2.2 问题定义

(1) 云任务集合

采用一维数组表示任务集合:

$$Task = \{task_1, task_2, \dots, task_m\} \quad (1)$$

式中, m 表示云任务的数量,每个任务都是相互独立的,执行过程中不可阻断.

(2) 虚拟机集合

采用一维数组表示虚拟机集合:

$$VmList = \{vm_1, vm_2, \dots, vm_n\} \quad (2)$$

式中, n 表示虚拟机的数量.

(3) 完成时间

任务完成时间:

$$total_time = \max(time_vm_i) \quad (3)$$

$$time_vm_i = \sum_{j=1}^k time_task_j \quad (4)$$

$$time_task_j = \frac{length_task_j}{mips_vm_i} \quad (5)$$

式中, $time_vm_i$ 表示第*i*个虚拟机的执行时间; $time_task_j$ 表示第*j*个任务的执行时间; $length_task_j$ 表示第*j*个任务的长度; $mips_vm_i$ 表示第*i*个虚拟机的处理速度.

(4) 完成成本

任务完成时间:

$$total_cost = \sum_{i=1}^n cost_vm_i \quad (6)$$

$$cost_vm_i = time_vm_i \times price_vm_i \quad (7)$$

式中, $cost_vm_i$ 表示第*i*个虚拟机的花费; $price_vm_i$ 表示第*i*个虚拟机的单价.

(5) 编码方式

本文采用一维数组进行编码, $\{num_1, num_2, \dots, num_m\}$,表示第0个任务在第 num_1 号主机上执行,第1号任务在第 num_2 号主机上执行,依次类推.

3 基于改进的乌鸦搜索算法的云任务调度算法

3.1 乌鸦搜索算法简介

乌鸦搜索算法的基本过程描述如下:在一个*D*维的空间中分布着一个群体大小为*M*的乌鸦群体,向量 $X^{i,iter}$ 表示 $Crow_i$ 在第*iter*次迭代中的位置,乌鸦*i*的位置更新分为两种状态:

一是当未发现被跟踪时, $Crow_j$ 藏食地会被 $Crow_i$ 发现,此时 $Crow_i$ 的位置更新为:

$$X^{i,iter+1} = X^{i,iter} + r_i \times fl \times (M^{j,iter} - X^{i,iter}) \quad (8)$$

二是当 $Crow_j$ 知道 $Crow_i$ 追踪时, $Crow_j$ 随意飞行来欺骗 $Crow_i$,此时 $Crow_i$ 的位置更新为空间内的一个随机位置:

$$X^{i,iter+1} = a \text{ random position} \quad (9)$$

向量 $M^{i,iter}$ 表示 $Crow_j$ 的藏食地,表示 $Crow_j$ 在第*iter*次迭代过程中的记忆位置,也是最优解. $M^{i,iter}$ 更新公式为:

$$M^{i, iter+1} = \begin{cases} X^{i, iter+1}, & f(X^{i, iter+1}) > f(M^{i, iter}) \\ M^{i, iter}, & \text{其他} \end{cases} \quad (10)$$

详细流程如下:

Step1:初始化参数:步长 fl 、感知概率 AP 、迭代次数 $iteration$ 、群体规模 N 、问题维度 D ;
 step2:初始化乌鸦的位置 X^i 和记忆 M^i ;
 step3:评估乌鸦的位置;
 step4:在乌鸦群体中随机选取 $Crow_j$, 如果 $r_i \geq AP$ 则 $Crow_j$ 按照公式(8)更新位置;否则, $Crow_j$ 的位置按照公式(9)更新;
 step5:检查新位置的可行性;
 step6:评估新位置,更新记忆;
 step7:判断当前的循环迭代次数是否满足条件,若不满足,返回至 step4,否则,迭代终止,输出当前最优解。

3.2 改进的乌鸦搜索算法

3.2.1 采用反向学习的初始化群体

原始的乌鸦搜索算法在初始化群体时,采用的是随机初始化的方式,初始解的随机性会导致分布不佳,会影响实验的收敛速度. 反向学习^[11] (OBL)的主要思想是在解空间内考虑原始解决方案的“对立面”,可以增加解空间的覆盖范围,从而找到更好的解决方案,从而提高收敛速度、准确度。

假设 CSA 随机生成的 $Crow_j$ 的位置为 $X_j(x_1, x_2, \dots, x_d)$, 则其反向位置为 $OBLX_j(x'_1, x'_2, \dots, x'_d)$ 满足:

$$x'_i = upper + lower - x_i \quad (11)$$

式中, $upper$ 为解空间的上界; $lower$ 为解空间的下界。

3.2.2 择优选取领导者

在 CSA 每次迭代过程中,被跟踪的乌鸦 $Crow_j$ 的选择是随机进行选择,这样的选择具有盲目性,使得 $Crow_j$ 的位置可能会远离最优解,而具有较高适应度值的乌鸦应该相对来说具有更好的可选中性。

本文采用的是一种基于适应度值的选择策略,在选择被跟踪 $Crow_j$ 时,在群体规模 M 下,随机选择 $M/2$ 只乌鸦,进行比较适应度值,选择适应度最佳的乌鸦作为被跟踪 $Crow_j$, 随机操作有利于搜索的探索性,择优有利于加速收敛。

3.2.3 乌鸦位置变异交叉

受遗传算法交叉变异操作启发,在迭代过程中,当乌鸦处于第一种更新状态时,当 $crow_i$ 位置更新完毕后,对新位置进行反向学习变异,其变异公式

如下:

$$x_{new_mut}[i][j] = (upper + lower - x_{new}[i][j]) \quad (12)$$

变异后得到的 $x_{new_mut}[i][j]$ 与 $x_{new}[i][j]$ 进行交叉重组,交叉重组公式如下:

$$x_{new_sec}[i][j] = x_{new}[i][j] + \omega \times x_{new_mut}[i][j] \quad (13)$$

式中, ω 为比例因子,每一个个体要选择一个新的 ω 值, ω 值计算如下:

$$\omega = -0.25 + 1.5 \times rand(0, 1) \quad (14)$$

3.3 算法流程描述

算法步骤如下:

Step1:初始化参数:步长 fl 、感知概率 AP 、迭代次数 $iteration$ 、群体规模 N 、问题维度 D ;

step2:采用反向学习初始化乌鸦的位置 X^i 和记忆 M^i ;

step3:评估乌鸦的位置;

step4:在乌鸦群体中择优选取 $Crow_j$, 如果 $r_i \geq AP$ 则 $Crow_i$ 按照公式(8)、(12)、(13)更新,然后根据是适应度值进行位置保留;否则, $Crow_i$ 的位置在可行区域内随机更新;

step5:检查新位置的可行性;

step6:评估新位置,更新记忆;

step7:判断当前的循环迭代次数是否满足条件,若不满足,返回至 step4,否则,迭代终止,输出当前最优解。

4 实验仿真分析

为了验证算法的有效性,先在标准测试函数中与 CSA 作对比,然后在 CloudSim 仿真平台下,与 GA、PSO、Min_Min、CSA 算法进行了任务调度的实验对比。

4.1 标准测试函数的对比与分析

为了验证 IMCSA 的算法性能,和原始 CSA 做了 6 组基准测试函数的实验,设置参数:种群大小 $M=30$, $AP=0.1$, $fl=2$, 最大迭代次数 1000. 每种算法独立运行 20 次,表 2 为对基准函数的寻优结果。

表 1 中列举的测试函数,包含单峰、多峰等不同特征的测试函数,单峰函数在给定的求解空间内只有一个严格的极大值或者极小值,因此单峰函数能够用来检验算法的收敛性能. 多峰函数在给定的求解空间内包含若干局部最优或全局最优解,因此多峰函数可以检验算法的探索性能。

表 1 基准测试函数			
函数名	维数	范围	最优值
Sphere	30	$[-100,100]$	0
Schwefel2.22	30	$[-10,10]$	0
Sum squares	30	$[-10,10]$	0
Ackley	30	$[-32,32]$	0
Rastrigin	30	$[5.12,5.12]$	0
Griewank	30	$[-600,600]$	0

从表 2 中可以看出,本文提出的算法(IMCSA)与乌鸦搜索算法(CSA)相比,对 Sphere 等六个函数都有较好的最优值、最差值、平均值以及标准差,在

表 2 IMCSA 和 CSA 对基准函数的寻优结果

函数	算法	最优值	最差值	平均值	标准差
Sphere	CSA	2.71E-02	2.24E-01	7.86E-02	4.78E-02
	IMCSA	5.21E-102	9.53E-84	5.39E-85	2.08E-84
Schwefel2.22	CSA	6.49E-01	3.79E+00	1.66E+00	8.61E-01
	IMCSA	1.55E-48	9.51E-42	5.12E-43	2.07E-42
Sum-Squares	CSA	1.88E-01	3.37E+00	1.10E+00	9.58E-01
	IMCSA	3.91E-102	1.61E-84	8.07E-86	3.52E-85
Ackley	CSA	2.67E+00	4.78E+00	3.61E+00	6.38E-01
	IMCSA	8.88E-16	8.88E-16	8.88E-16	0
Rastrigin	CSA	7.99E+00	3.98E+01	2.39E+01	8.03E+00
	IMCSA	0	0	0	0
Griewank	CSA	2.74E-03	2.85E-02	1.31E-02	7.80E-03
	IMCSA	0	0	0	0

4.3 算法参数设置

对比算法参数设置如下表所示

表 3 参数设置	
算法	参数值
CSA	步长 $fl=2$ 、感知概率 $AP=0.5$
PSO	加速常数 $c1=c2=2$ 、惯性权重 $w=1.4$ 、最大速度 $vmax=2$
GA	交叉率 0.85 变异率 0.1
本文算法	步长 $fl=2$ 、感知概率 $AP=0.5$

4.4 实验结果分析

从图 1 可以看出,当任务数较小的时候,5 种算法的差异性相对来说较小,随着任务数量的增多,CSA 和 IMCSA 的任务执行的完成时间明显优于 PSO、GA、Min_Min 算法,优势越来越明显,IMCSA 算法的任务完成时间也优于原始 CSA 算法。

五种算法的任务执行成本中,Min_Min 算法的任务执行成本最高,CSA 算法和 PSO 算法的执行成本差异不明显,IMCSA 的任务执行成最低。当任

单峰函数的测试效果上,具有很快的收敛速度,在多峰函数的测试上,更具有探索开发性能。

4.2 实验环境与参数设置

采用 CloudSim 云仿真平台,实验中用到的对比算法有原始的乌鸦搜索算法(CSA)、粒子群(PSO)算法、遗传算法(GA)、Min_Min 算法,通过对任务完成时间、执行成本的分析,来验证本文算法的性能与效果。设置任务长度为 10 000—40 000MIPS,任务数 100—500,虚拟机资源数 30,虚拟机处理速度 200—1 000 MIPS,迭代次数为 200,群体规模 M 为 50。为避免随机性,每个算法独立运行 20 次取平均值。

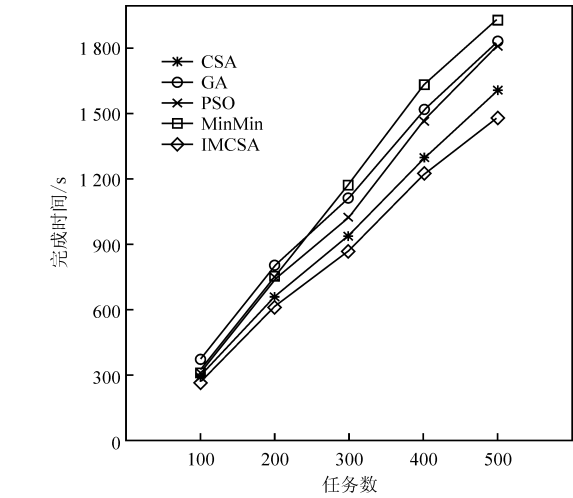


图 1 任务完成时间对比图

务数较少时,IMCSA 算法在执行成本上略优于 CSA、GA、PSO 和 Min_Min 算法,并且随着任务数的增加,优势越来越明显。

5 结束语

本文研究了改进乌鸦搜索算法(IMCSA)的云任务调度,建立了以任务的执行时间和成本为目标的调度模型,并与传统的基本算法进行了对比,仿真实验表明,IMCSA 在性能上优于粒子群算法(PSO)、遗传算法(GA)、Min_Min 和 CSA 算法.在实际的云环境中,云计算的任务调度影响因素还有很多,在未来的研究中将考虑虚拟资源的能耗、Qos 等因素.

参考文献:

- [1] 齐平,王福成,王必晴,等.云计算环境下基于可靠性感知的任务调度算法[J].计算机工程与科学,2018,40(11):1925-1935.
- [2] GABI D, ISMAIL A S, ZAINAL A, et al. Orthogonal Taguchi-based cat algorithm for solving task scheduling problem in cloud computing[J]. Neural Computing and Applications, 2018, 30(6): 1845-1863.
- [3] 萨日娜.基于蚁群粒子群优化算法的云计算资源调度方案[J].吉林大学学报(理学版), 2017, 55(6): 1518-1522.
- [4] 张焕青,张学平,王海涛,等.基于负载均衡蚁群优化算法的云计算任务调度[J].微电子学与计算机, 2015, 32(5): 31-35.
- [5] ASKARZADEH A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm[J]. Computers & Structures, 2016, 169: 1-12.
- [6] SAYED G I, HASSANIEN A E, AZAR A T. Feature selection via a novel chaotic crow search algorithm

[J]. Neural Computing and Applications, 2019, 31(1): 171-188.

- [7] MOHAMMADI F, ABDI H. A modified crow search algorithm (MCSA) for solving economic load dispatch problem[J]. Applied Soft Computing, 2018, 71: 51-65.
- [8] DiAZ P, PÉREZ-CISNEROS M, CUEVAS E, et al. An improved crow search algorithm applied to energy problems[J]. Energies, 2018, 11(3): 571.
- [9] 刘雪静,贺毅朝,路凤佳,等.基于 Lévy 飞行的差分乌鸦算法求解折扣 {0-1} 背包问题[J]. 计算机应用, 2018, 38(2): 433-442.
- [10] SREENU K, SREELATHA M. W-Scheduler: whale optimization for task scheduling in cloud computing[J]. Cluster Computing, 2019, 22: 1087-1098.
- [11] TIZHOOSH H R. Opposition-Based Learning: A New Scheme for Machine Intelligence[C], International Conference on Computational Intelligence for Modeling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, IEEE, 2005, 695-701.

作者简介:

林涛 男, (1970-), 博士, 教授. 研究方向为云计算、物联网关键技术、智能控制.

郝章肖(通讯作者) 男, (1992-), 硕士研究生. 研究方向为云计算, 算法优化. E-mail: 1589723192@qq.com

冯竞凯 男, (1994-), 硕士研究生. 研究方向云计算、容器技术.