

DOI: 10.7652/xjtuxb201410019

AAFSA-RA: 一种采用高级人工鱼群算法的多资源分配方法

桓自强^{1,2}, 倪宏¹, 胡琳琳¹, 郭志川¹

(1. 中国科学院声学研究所国家网络新媒体工程技术研究中心, 100190, 北京; 2. 中国科学院大学, 100049, 北京)

摘要: 针对资源受限环境中多任务间的多资源分配问题,提出一种采用高级人工鱼群算法的多资源分配方法(AAFSA-RA)。对多资源分配进行建模,结果表明多资源分配是多约束多维度的复杂函数最优化问题。针对基本人工鱼群算法(AFSA)在寻优过程中速度慢的问题,提出一种高级人工鱼群算法(AAFSA)。通过适当地限制人工鱼的随机游动来改进觅食行为。引入步长和视野的两级动态调整机制,基于鱼群进化过程进行一级调整,基于人工鱼个体的食物浓度及变化情况进行二级调整。根据人工鱼个体食物浓度控制人工鱼的生命周期。基于 AAFSA 进行多资源分配。仿真实验结果表明:AAFSA-RA 可以稳定地搜索到全局最优值,与现有 AFSA 算法相比,AAFSA 寻优速度更快;与传统资源分配方法相比,AAFSA-RA 可以达到更高的精度,并且对效用函数无特殊要求。

关键词: 多资源分配;服务质量;效用函数;高级人工鱼群算法;最优化

中图分类号: TP316 **文献标志码:** A **文章编号:** 0253-987X(2014)10-0120-06

AAFSA-RA: A Multi-Resource Allocation Method Based on an Advanced Artificial Fish Swarm Algorithm

HUAN Ziqiang^{1,2}, NI Hong¹, HU Linlin¹, GUO Zhichuan¹

(1. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: A muti-resource allocation method based on an advanced artificial fish swarm algorithm (AAFSA-RA) is proposed to solve the problem of multi-resource allocation among tasks in a resource-constrained environment. The muti-resource allocation is modeled, and is derived as a complex optimization problem with muti-constraints. Then an advanced artificial fish swarm algorithm (AAFSA) is proposed to improve the computational speed of the basic artificial fish swarm algorithm (AFSA). Random walks of artificial fishes are properly limited to improve prey behavior. A two-stage dynamic adjustment mechanism is introduced. The adjustment in the first-stage is based on the evolutionary process of fish school, and the adjustment in the second-stage is based on the food concentration of single artificial fish and its changes. The life cycle of each artificial fish is controlled according to its food concentration. Finally, the problem of muti-resource allocation is solved using AAFSA. Simulation results show that AAFSA-RA steadily finds the global optimum value with a speed faster than the speed of existing algorithms based on

收稿日期: 2014-05-23。 作者简介: 桓自强(1988—),男,博士生;倪宏(通信作者),男,研究员,博士生导师。 基金项目: 国家科技支撑计划资助项目(2012BAH73F01);中国科学院战略性先导科技专项资助项目(XDA06040501)。

网络出版时间: 2014-09-23 网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20140923.1748.005.html>

AFSA; Comparisons with the traditional resource allocation methods show that the AAFSA-RA achieves higher precision without any special requirements on utility functions.

Keywords: multi-resource allocation; quality of service; utility function; advanced artificial fish swarm algorithm; optimization

在资源受限环境例如无线传感器网络^[1]、智能终端^[2]、无线电系统^[3]和云计算^[4]等应用场景中,软件的复杂性与系统资源的有限性使多任务间存在资源竞争,因此多任务间多共享资源的分配问题都是研究的热点。资源分配问题可分为两类:以系统效用最大化为目标和以应用间效用公平性为目标^[2]。在以用户体验为核心的今天,如何使得系统整体效用最优显得尤为重要,也是本文讨论的重点。

Rajkumar 等在解决多资源分配问题时,通过引入系统效用与资源具有线性关系,从而将问题转化为混合整数规划问题进行求解,但线性关系的假设限制了可用性^[5]。Lee 等在混合整数规划求解中引入了本地搜索技术,从而使得求解速度实现两个数量级的提升,达到近似解的 95%^[6]。Khan 等将多媒体系统的系统效用最大化问题转化为多维多选择背包问题,并给出了分支界限法^[7]。

对于多任务间多资源分配问题,传统的寻优方法因为任务数量和资源数量的增加,存在着算法复杂、收敛速度慢等问题。启发式算法在寻优中优势明显^[8]。Khan 等通过构造贪心启发函数提出一种启发式解法 HEU,但得到的近似解距离最优解较远^[7];Mostofa 等基于凸包的性质来提出一种 C-HEU 启发式算法,但是也只能达到最优解的 90%^[9]。经济学中的拍卖算法在无线电系统的资源分配中取得了良好效果^[3]。陈俊杰等通过假设资源消耗函数为凸函数将多资源分配问题转化为凸优化问题,并通过经济学中的多标拍卖算法来求解最优值,但是凸函数的假设限制了适用范围^[2]。

人工鱼群算法(artificial fish swarm algorithm, AFSA)是一种通过模拟鱼群行为、基于动物自治体的启发式的寻优方法^[10]。AFSA 自提出以来已在控制、网络、图形处理、数据挖掘、人工神经网络、调度和信号处理等工程领域取得了重要应用^[11]。本文通过对基本 AFSA 算法进行改进,提出一种高级人工鱼群算法(advanced artificial fish swarm algorithm, AAFSA),并进一步提出了采用高级人工鱼群算法的多资源分配方法(advanced artificial fish swarm algorithm based multi-resource allocation, AAFSA-RA)。

1 多资源分配模型

1.1 资源和任务集

对于资源受限环境,并行运行的 n 个任务的集合为 $\{\pi_1, \pi_2, \dots, \pi_n\}$,存在 m 种共享资源,例如终端系统的内存、CPU 带宽、磁盘 IO,或无线传感器网络中的带宽资源等,资源集合为 $\{R_1, R_2, \dots, R_m\}$ 。对于任一种资源 R_i ,其由 n 个任务共享,对于任务 π_j 获得的资源为 $R_{ij} (\in [0, R_i])$,存在约束关系

$$R_{i1} + R_{i2} + \dots + R_{in} = \sum_{j=1}^n R_{ij} \leq R_i \quad (1)$$

任务 π_j 获得的资源集合为 $\{R_{1j}, R_{2j}, \dots, R_{mj}\}$,给出任务 π_j 的效用函数的定义为

$$q_j'' = f_j(R_{1j}, R_{2j}, \dots, R_{mj}) \quad (2)$$

效用函数需要满足非精确计算模型,即任务 π_j 对资源 R_i 的需求,存在最低资源需求 $R_{ij}^{\min}$,这部分强制执行,保证任务最低的服务质量 QoS 需求。增加资源可以提高 QoS,这部分可选执行,但是存在需求上限 $R_{ij}^{\max}$,资源达到上限后再次增加资源并不能提高 QoS 水平。

对于不同的任务,在系统中存在不同的重要性,重要性可以由系统根据任务类型预设,也可以由用户手动指定。因此对不同的应用,需要引入权值 ω_j ,则效用函数转化为

$$q_j' = \omega_j q_j'' = \omega_j f_j(R_{1j}, R_{2j}, \dots, R_{mj}) \quad (3)$$

式(3)需要满足如下条件:

(1)存在资源需求限制

$$R_{ij}^{\min} \leq R_{ij} \leq R_{ij}^{\max}, \quad 1 \leq i \leq m \quad (4)$$

(2)效用边界

$$\begin{aligned} q_j'(R_{1j}^{\min}, R_{2j}^{\min}, \dots, R_{mj}^{\min}) &= q_{j\min}' \\ q_j'(R_{1j}^{\max}, R_{2j}^{\max}, \dots, R_{mj}^{\max}) &= q_{j\max}' \end{aligned} \quad (5)$$

系统中的任务很多,不同的任务可能有不同的效用函数,为了对不同任务的效用有个统一的衡量标准,对服务质量进行归一化,归一化的效用函数为

$$Q_j(R_{1j}, \dots, R_{mj}) = \begin{cases} 0, & R_{ij} = R_{ij}^{\min} \\ \frac{q_j' - q_{j\min}'}{q_{j\max}' - q_{j\min}'}, & R_{ij}^{\min} < R_{ij} < R_{ij}^{\max} \\ 1, & R_{ij} = R_{ij}^{\max} \end{cases} \quad (6)$$

对于式(6),存在如下条件:①对于 m 种资源、 n 个任务的系统来说, $1 \leq i \leq m, 1 \leq j \leq n$; ②归一化后,存在 $0 \leq Q_j \leq 1$ 。

1.2 系统效用最大化模型

根据上面的分析,下面给出具有 n 个任务、 m 种资源的系统效用最大化模型。

定义 1 对于资源集 $\{R_1, R_2, \dots, R_m\}$ 和任务集 $\{\pi_1, \pi_2, \dots, \pi_n\}$ 的系统来说,任务 π_j 的效用是 $Q_j(R_{1j}, \dots, R_{mj})$, 需寻求一组 $m \times n$ 维度的资源配置矩阵 U_{ij} , 使得下式在资源约束条件下成立

$$\max \sum_{j=1}^n Q_j(R_{1j}, \dots, R_{mj}) \quad (7)$$

需满足以下约束条件:① Q_j 满足上述式(6)的归一化表述;②对于资源 $R_i (1 \leq i \leq m)$, 存在资源限制

$$\sum_{j=1}^n R_{ij} \leq R_i, \quad 1 \leq i \leq m \quad (8)$$

③对于任务 $\pi_j (1 \leq j \leq n)$, 其对每种资源的需求存在上下界

$$R_{ij}^{\min} < R_{ij} < R_{ij}^{\max}, \quad 1 \leq i \leq m \quad (9)$$

2 高级人工鱼群算法

通过分析基本人工鱼群算法存在的问题,综合考虑寻优精度和搜索速度等因素,本文提出一种高级人工鱼群算法 AAFSA。

2.1 现有人工鱼群算法

AFSA 算法^[10]作为一种群体智能算法,具有并行性、快速性和简单性等特点,但是其在很多工程领域也存在着时间复杂度高、收敛精度低以及局部搜索和全局搜索不能很好平衡等问题。

针对 AFSA 的不足,研究者进行了广泛的改进研究。Jiang 等对人工鱼的个体行为引入全局最优状态的参考,在觅食、聚群和追尾等行为中综合考虑当前局部搜索值和全局最优值^[12]; Gao 等引入交叉算子来促进人工鱼的多样化,继承父辈属性,同时引入文化算法优化盲目搜索^[13];王联国等提出全局人工鱼群算法(global edition artificial fish swarm algorithm, GAFSA),一是引入步长和视野的动态调整,二是在聚群和追尾行为中使用全局最优信息等^[14]。目前的改进算法在低维数、无约束或少约束的环境下,都对 AFSA 有不同程度的改善,尤其以 GAFSA 收敛速度最快。在多资源分配环境中,目标函数是复杂的高纬度函数,并且存在多种约束,需要一种速度更快、精度更高的算法。

2.2 高级人工鱼群算法

2.2.1 公告板 公告板用来记录全局属性,负责记

录全局最大值,以及最大值对应的变量。

2.2.2 步长和视野调整 人工鱼群算法的进化过程是一个逐步趋于极值的过程,寻优初期需要较大的步长和视野来进行大范围探索以提高速度,而寻优后期需要较小的步长和视野来进行精度提升。文献[14]正是基于此提出了如下公式

$$\left. \begin{aligned} V &= V_{\alpha} + V_{\min} \\ S &= S_{\alpha} + S_{\min} \\ a &= \exp(-30(t/T_{\max})^s) \end{aligned} \right\} \quad (10)$$

式(10)可以实现步长的动态调整,并逐渐递减,但是递减速度单纯通过这个公式并不能有效控制,递减过快容易导致前期搜索阶段变短,递减过慢又会使后期精度得不到保证。同时,后期鱼群的高精度搜索,只需要由部分高浓度的人工鱼执行,这也需要对不同人工鱼区别对待。因此,本文 AAFSA 通过引入两级调整策略来有效地解决这个问题。

一级调整策略如式(11),其中变量 i 为鱼群进化代数,变量 k 为衰减因子,用来控制一级步长和视野的衰减速度。与式(10)相比,避免了需要指定最大迭代次数 $T_{\max}$ 的问题。

$$\left. \begin{aligned} \alpha &= 0.95^{(i/k)} \\ S &= S_{\alpha} + S_{\min} \\ V &= V_{\alpha} + V_{\min} \end{aligned} \right\} \quad (11)$$

二级调整策略的原则是,当高浓度人工鱼(例如大于 95% 最大值)连续几次未搜寻到更优值时,进行视野和步长调整。为此,对每条人工鱼引入个体步长 s_f 、个体视野 v_f 和个体最优值未变化次数 w_f 等属性。调整策略为

$$\left. \begin{aligned} s_f(i) &= S0.6^{w_f(i)/t} + 0.1S_{\min} \\ v_f(i) &= V0.6^{w_f(i)/t} + 0.1V_{\min} \end{aligned} \right\} \quad (12)$$

式中: i 表示人工鱼序号; t 表示未变化次数的阈值,一般可取 2~4。另外公式只适用高浓度人工鱼,例如食物浓度大于 95% 最大值。

2.2.3 生命周期 定期对鱼群中的低食物浓度个体进行重新初始化。一种可行策略是:每隔一定次数 N , 以食物浓度对鱼群排序,对后 5% 的个体进行重新初始化。

2.2.4 觅食行为 AAFSA-Prey 基本的觅食行为中,会在邻域内执行随机行为,寻到更优值则向其随机移动一步,否则继续随机搜索,反复尝试 n_{try} 次,若还未发现更优值,则随机移动一步。为了限制过分的随机行为,采取两种方式:一是对找到的更优值,直接一步移动到更优值的位置;二是对觅食阶段未找到更优值的高浓度人工鱼,采取原地不动的策略,

避免随机行为导致其偏离高浓度区域。

2.2.5 聚群行为 AAFSA-Swarm 邻域中心位置 X_c , 邻域鱼群个数 n_f , 拥挤因子 δ , 在 $Y_c/n_f > \delta Y_i$ 时, 可以执行聚群动作

$$X_i^{t+1} = X_i^t + ((X_c - X_i^t) / \|X_c - X_i^t\|) s_f(i) \text{rand}(0, 1) \quad (13)$$

为了加速聚群行为, 实际应用中可以将邻域范围扩展为整个鱼群。

2.2.6 追尾行为 AAFSA-Follow 邻域最优位置 X_{best} , X_{best} 的邻域鱼群个数 n_f , 拥挤因子 δ , 在 $Y_{\text{best}}/n_f > \delta Y_i$ 时, 执行追尾行为

$$X_i^{t+1} = X_i^t + ((X_{\text{best}} - X_i^t) / \|X_{\text{best}} - X_i^t\|) s_f(i) \text{rand}(0, 1) \quad (14)$$

为了加速追尾行为, 实际应用中可以将邻域范围扩展为鱼群的整体。

2.2.7 随机行为 AAFSA-Rand 人工鱼个体在视野内随机行进的动作, 也是觅食行为的缺省动作

$$X_i^{t+1} = X_i^t + \text{rand}(v_f(i)) \quad (15)$$

3 采用 AAFSA 的资源分配算法

本节将前面提出的 AAFSA 算法运用到多资源分配问题中。式(7)的寻优可以等效为

$$F(x_1, x_2, \dots, x_{m \times n}) = \max \sum_{j=1}^n Q_j(R_{1j}, \dots, R_{mj}) \quad (16)$$

人工鱼个体 fish 表示为 $\langle x_1, x_2, \dots, x_{m \times n} \rangle$, 其食物浓度函数如式(16)所示。元素映射关系为

$$x_{(i-1) \times n + j} = R_{ij} \quad (17)$$

在多约束如式(8)、(9)的情况下, AAFSA 算法在执行随机行为时需要考虑约束, 同时在觅食、聚群和追尾等行为执行后, 都需要对新的人工鱼进行基于约束的合理化调整, 以消除无效值。

3.1 基于约束的随机行为

式(15)的随机行为, 在多资源分配问题中需要考虑其前后可移动距离的有效性。

RA-Rand Algorithm

input: fish($x_1, x_2, \dots, x_{m \times n}$), fish_visual

output: rand_step

for $i=1$ to m

for $j=1$ to n

left = max($-(x_{(i-1) \times n + j} - R_{ij}^{\min})$, -fish_visual)

right = min($R_{ij}^{\max} - x_{(i-1) \times n + j}$, fish_visual)

rand_step $_{(i-1) \times n + j}$ = left + rand(1) × (right - left)

3.2 基于约束的人工鱼调整

多约束的环境使得寻优过程容易造成人工鱼越界, 用基于约束的人工鱼调整来保证每次迭代中人工鱼都是合理的。

RA-Adjustfish Algorithm

input: old_fish($x_1, x_2, \dots, x_{m \times n}$)

output: new_fish($nx_1, nx_2, \dots, nx_{m \times n}$)

new_fish = old_fish

for $i=1$ to m

for $j=1$ to n

if $nx_{(i-1) \times n + j} > R_{ij}^{\max}$ $nx_{(i-1) \times n + j} = R_{ij}^{\max}$

if $nx_{(i-1) \times n + j} < R_{ij}^{\min}$ $nx_{(i-1) \times n + j} = R_{ij}^{\min}$

for $i=1$ to m

sum = $\sum_{j=1}^n (nx_{(i-1) \times n + j} - R_{ij}^{\min})$

resfree = $R_i - \sum_{j=1}^n R_{ij}^{\min}$

for $j=1$ to n

$nx_{(i-1) \times n + j} = R_{ij}^{\min} + (nx_{(i-1) \times n + j} - R_{ij}^{\min}) \times (\text{resfree}/\text{sum})$

if $nx_{(i-1) \times n + j} > R_{ij}^{\max}$ $nx_{(i-1) \times n + j} = R_{ij}^{\max}$

3.3 AAFSA-RA 算法流程

步骤 1 初始化鱼群参数: 鱼群个体数、视野 V 和步长 S 的初始值, 一级、二级调整迭代最小值等, 觅食行为的 n_{try} 次数等。初始化个体参数: 步长 s_f 、视野 v_f 、浓度未变化次数 w_f 等。

步骤 2 根据式(11)进行视野和步长的一级调整, 根据当前鱼群进化次数, 依据 2.2.3 节进行鱼群生命周期调整。

步骤 3 对于鱼群进行遍历, 取出此次进化中还未处理的个体, 执行步骤 4, 否则执行步骤 7。

步骤 4 依据 2.2.2 节判断是否满足二级调整的条件, 根据式(12)进行步长和视野的二级调整, 否则, 使用目前的一级步长和视野。

步骤 5 对人工鱼执行觅食, 若得到比公告板的值更大的值, 则执行步骤 6; 否则执行聚群, 若得到比公告板的值更大的值, 则执行步骤 6; 否则执行追尾, 若得到比公告板的值更大的值, 则执行步骤 6; 否则, 取 3 种行为的最大值为该人工鱼的新值。

步骤 6 判断此次迭代中第 i 条人工鱼的食物浓度是否变化, 更新 w_f 的值, 转向步骤 3。

步骤 7 判断鱼群迭代次数是否达到, 如果达到则终止进化过程; 否则转向步骤 2。

4 实验与结果分析

为了验证本文提出 AAFSA-RA 的有效性, 选

取共享资源种类 $m=4$, 并行任务数为 $n=10$ 。对于任务效用函数, 参考文献[15]给出的资源效用函数类别, 给出多资源情况下 4 种任务类型:

(1)
$$Q_j = (\sum_{i=1}^4 R_{ij} - \sum_{i=1}^4 R_{ij}^{\min}) / (\sum_{i=1}^4 R_{ij}^{\max} - \sum_{i=1}^4 R_{ij}^{\min})$$

(2)
$$Q_j = (\prod_{i=1}^4 R_{ij} - \prod_{i=1}^4 R_{ij}^{\min}) / (\prod_{i=1}^4 R_{ij} - \prod_{i=1}^4 R_{ij}^{\min})$$

(3)
$$Q_j = \sin(\pi/2) (\sum_{i=1}^4 R_{ij} - \sum_{i=1}^4 R_{ij}^{\min}) / (\sum_{i=1}^4 R_{ij}^{\max} - \sum_{i=1}^4 R_{ij}^{\min})$$

(4)
$$Q_j = 1 - \sin(\pi/2) (\prod_{i=1}^4 R_{ij}^{\max} - \prod_{i=1}^4 R_{ij}) / (\prod_{i=1}^4 R_{ij}^{\max} - \prod_{i=1}^4 R_{ij}^{\min})$$

上面 4 种任务类型均为经过式(6)归一化之后的形式, 属性的取值范围均为 $0 \leq Q_j \leq 1, 1 \leq i \leq 4, 1 \leq j \leq 10$ 。4 种资源的总量均为 1, 10 个任务的资源需求上下界见表 1。

表 1 任务类型及最小最大资源需求

任务编号	最小需求	最大需求	任务类型
1	(0.02, 0.01, 0.02, 0.01)	(0.35, 0.4, 0.2, 0.3)	(4)
2	(0.01, 0.01, 0.01, 0.01)	(0.3, 0.25, 0.22, 0.25)	(1)
3	(0.02, 0.03, 0.03, 0.01)	(0.25, 0.3, 0.35, 0.2)	(4)
4	(0.01, 0.01, 0.02, 0.03)	(0.2, 0.3, 0.3, 0.3)	(2)
5	(0.02, 0.03, 0.01, 0.03)	(0.32, 0.3, 0.2, 0.2)	(1)
6	(0.03, 0.02, 0.03, 0.01)	(0.28, 0.22, 0.38, 0.19)	(1)
7	(0.01, 0.01, 0.02, 0.02)	(0.19, 0.18, 0.23, 0.26)	(3)
8	(0.01, 0.01, 0.01, 0.01)	(0.32, 0.28, 0.19, 0.18)	(3)
9	(0.02, 0.03, 0.01, 0.02)	(0.38, 0.3, 0.32, 0.29)	(2)
10	(0.01, 0.02, 0.01, 0.02)	(0.3, 0.4, 0.5, 0.35)	(2)

针对上述资源分配问题, 分别比较文献[10]中的基本算法 AFSA、文献[14]的改进算法 GAFSA、以及本文所提出的改进算法 AAFSA 的寻优效果。3 种算法中有共同的参数。鱼群数目为 50 条, 觅食阶段尝试次数 n_{try} 为 20 次, 拥挤因子为 0.75。对于 AFSA-RA, 选取步长 S 为 0.002, 视野 V 为 0.01; 对于 GAFSA-RA, 初始步长 S 和 V 分别为 0.02 和 0.1, $S_{\min}=0.000\ 2, V_{\min}=0.001$, 式(10)中的参数 s 取 3; 对于 AAFSA-RA, 初始 S, V 分别为 0.02、0.1, 其在步长和视野的一级调整式(11)中的衰减因子 $k=15$, 二级调整式(12)的阈值 $t=2, S_{\min}=0.000\ 2, V_{\min}=0.001$, 生命周期调整周期为 25 次(见 2.2.2 节)。

用相同的初值对 50 条鱼进行初始化, 3 种方法的寻优效果如图 1 所示, 可见本文提出的 AAFSA-RA 算法具有非常明显的优势。AFSA-RA 固定步长和视野, 在精细调整参数的情况下, 使得前 200 次进化保持一定的寻优速度, 但随着寻优进入后期, 固定的步长和视野导致寻优速度较慢(迭代到 300 次也只能收敛到 3.8 左右)。GAFSA-RA 采用动态视野和步长, 前期搜索阶段因无法对部分高浓度区域的个体进行精细搜索导致前期搜索效率低, 后期搜索的视野和步长递减过快也导致高浓度区域外的其他个体进化效果减弱, 从而整体寻优比较慢。AAFSA 算法通过两级视野和步长的动态调整, 并对不同浓度的鱼区分对待, 在整个进化过程中每条鱼均可以很好地平衡粗放搜索和精细搜索。

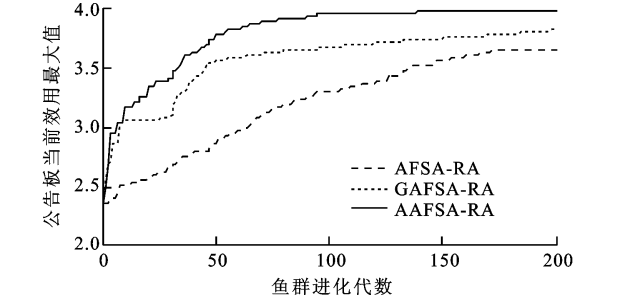


图 1 3 种算法寻优效果对比

为了更好地证明本文方法的有效性, 下面对每种方法各进行 20 次实验, 进化代数依然为 200 代, 但每次选取随机初值, 结果见表 2。

表 2 3 种算法 20 次实验结果对比

算法	配置	最大值	最小值	平均值	方差
AFSA-RA	a	3.335 3	3.212 8	3.277 6	0.001 2
AFSA-RA	b	3.704 8	3.623 5	3.666 8	0.000 5
AFSA-RA	c	3.416 4	3.283 8	3.345 3	0.001 4
GAFSA-RA	a	3.827 7	3.685 4	3.741 4	0.001 5
GAFSA-RA	b	3.860 9	3.773 8	3.819 7	0.000 6
GAFSA-RA	c	3.849 1	3.747 7	3.806 8	0.000 9
AAFSA-RA		3.980 8	3.960 8	3.973 1	0.000 02

在表 2 中, 对于 GAFSA-RA, 分别取 3 种配置(配置 a: $s=2$; b: $s=3$; c: $s=4$); 对于 AFSA-RA, 步长、视野分别取 3 种组合(a: $s=0.001, v=0.005$; b: $s=0.002, v=0.01$; c: $s=0.01, v=0.05$)。可以看出, AFSA-RA 和 GAFSA-RA 均在配置区间(a, c)中存在较好的配置 b。AAFSA-RA 算法与前面两种算法的最好配置相比, 总是可以更快地寻到比较好的结果, 并且 2×10^{-5} 的方差说明算法非常稳定。

表 3 AAFSA-RA 不同进化代数 20 次优化对比

进化代数	最大值	最小值	平均值	方差
100	3.950 0	3.921 8	3.934 5	0.000 060
500	3.995 0	3.980 4	3.988 2	0.000 017
1 000	3.997 4	3.987 0	3.990 9	0.000 008

进一步,AAFSA-RA 的精度可以随着迭代次数的增加而稳定地提高,见表 2。由表 2、表 3 可见,在只进化 100 代的情况下,可以平均收敛到最优值的 98%以上,进化 200 代可以收敛到 99%以上,精度明显优于文献[9]的精度。AAFSA-RA 在随机初值和不同进化代数的情况下的方差均比较小,也说明其寻优效果是比较稳定的。

5 结 论

本文主要研究了资源受限环境中多任务间的多资源分配问题,针对其多约束、多维度和复杂度高的特点,提出了一种采用高级人工鱼群算法的多资源分配方法 AAFSA-RA。AAFSA-RA 基于高效改进的人工鱼群算法 AAFSA,而 AAFSA 具有基本鱼群算法对目标函数无要求对初值不敏感以及求解过程高度并发的特点;同时通过引入步长和视野的二级调整机制,觅食中随机行为抑制和生命周期调控机制等提高了寻优速度。实验结果表明,AAFSA-RA 方法与现有鱼群算法相比收敛速度更快,其进化 100 代可收敛到最优值的 98%以上,进化 200 代可以收敛到 99%以上,在精度上也明显优于传统资源分配算法。

参考文献:

[1] CHITNIS M, PAGANO P, LIPARI G, et al. A survey on bandwidth resource allocation and scheduling in wireless sensor networks [C]// Proceedings of the International Conference on Network-Based Information Systems. Piscataway, NJ, USA: IEEE, 2009: 121-128.

[2] 陈俊杰,倪宏,孙鹏,等. 采用定价机制的多媒体系统多资源分配算法 [J]. 西安交通大学学报, 2012, 46(6): 98-103.

CHEN Junjie, NI Hong, SUN Peng, et al. Pricing mechanism based multi-resource allocation for multimedia system [J]. Journal of Xi'an Jiaotong University, 2012, 46(6): 98-103.

[3] ZHANG Yang, LEE C, NIYATO D, et al. Auction approaches for resource allocation in wireless systems;

a survey [J]. Communications Surveys and Tutorials, 2013, 15(3): 1020-1041.

[4] PARIKH S M. A survey on cloud computing resource allocation techniques [C] // Proceedings of the 2013 Nirma University International Conference on Engineering. Piscataway, NJ, USA: IEEE, 2013: 1-5.

[5] RAJKUMAR R, LEE C, LEHOCZKY J, et al. Practical solutions for QoS-based resource allocation problems [C] // Proceedings of the IEEE Real-Time Systems Symposium. Piscataway, NJ, USA: IEEE, 1998: 296-306.

[6] LEE C, LEHOCZKY J, SIEWIOREK D, et al. A scalable solution to the multi-resource QoS problem [C] // Proceedings of the IEEE Real-Time Systems Symposium. Piscataway, NJ, USA: IEEE, 1999: 315-326.

[7] KHAN M S. Quality adaptation in a multisession multimedia system: model, algorithms and architecture [D]. Victoria, BC, Canada: University of Victoria, 1998.

[8] JUEDES D, DREWS F, WELCH L, et al. Heuristic resource allocation algorithms for maximizing allowable workload in dynamic, distributed real-time systems [C] // Proceedings of the 18th International Conference on Parallel and Distributed Processing Symposium. Piscataway, NJ, USA: IEEE, 2004: 117-124.

[9] MOSTOFA A M, SOHEL R M, KAYKOBAD M, et al. Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls [J]. Computers and Operations Research, 2006, 33(5): 1259-1273.

[10] 李晓磊,邵之江,钱积新,等. 一种基于动物自治体的寻优模式: 鱼群算法 [J]. 系统工程理论与实践, 2002, 22(11): 32-38.

LI Xiaolei, SHAO Zhijiang, QIAN Jixin, et al. An optimizing method based on autonomous animats: fish-swarm algorithm [J]. Systems Engineering in Theory and Practice, 2002, 22(11): 32-38.

[11] NESHAT M, SEPIDNAM G, SARGOLZAEI M, et al. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications [EB/OL]. (2012-05-06) [2013-05-23]. <http://link.springer.com/article/10.1007/s10462-012-9342-2>.

[12] JIANG Mingyan, YUAN Dongfeng, CHENG Yongming. Improved artificial fish swarm algorithm [C] // Proceedings of the 5th International Conference on Natural Computation. Piscataway, NJ, USA: IEEE, 2009: 281-285.

(下转第 136 页)