

随机资源约束项目调度问题基于序的果蝇算法

郑晓龙, 王 凌[†]

(清华大学 自动化系, 北京 100084)

摘要: 针对项目活动工期为随机变量的资源约束项目调度问题, 提出一种基于序的果蝇算法. 为了实现随机环境下解的有效评价, 提出一种预选机制, 并采用基于序的最优计算量分配技术. 为了使果蝇算法能够求解资源约束项目调度问题, 采用交换操作执行果蝇算法的嗅觉搜索, 并采用保优更新操作执行视觉搜索. 为了均衡算法的局部搜索和全局搜索能力, 在标准果蝇算法中引入了协作进化环节并采用两点交叉操作加以实现. 在不同随机分布的情况下, 采用标准测试集进行仿真测试. 与现有算法的比较结果验证了所提预选机制和基于序的果蝇算法的有效性.

关键词: 随机资源约束项目调度; 果蝇算法; 协作进化; 预选机制; 最优计算量分配

中图分类号: TP273

文献标识码: A

An order-based fruit fly optimization algorithm for stochastic resource-constrained project scheduling

ZHENG Xiao-long, WANG Ling[†]

(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: An order-based fruit fly optimization algorithm (OFOA) is proposed to solve the resource-constrained project scheduling problem (RCPSP) with stochastic activity duration. To evaluate solutions effectively under the stochastic environment, a pre-select scheme is proposed, and the order-based optimal computing budget allocation (OCBA) is adopted. To make the FOA suitable for RCPSP, the swap operator is used to perform the smell-based search, and the elite updating operator is used as vision-based search. To balance the exploration and exploitation abilities, a collaborative search element is embedded into the original FOA and implemented by using the two-point crossover operator. Simulation tests are carried out with the benchmark dataset by taking several types of distributions into account. The comparisons with existing algorithms demonstrate the effectiveness of the proposed pre-select scheme and the OFOA.

Key words: stochastic resource-constrained project scheduling; fruit fly optimization algorithm; collaborative search; pre-select scheme; optimal computing budget allocation

1 引言(Introduction)

资源约束项目调度问题(resource-constrained project scheduling problem, RCPSP)是在满足项目活动的优先约束以及资源的供给约束的前提下, 合理分配资源和安排项目活动的开始时间, 以实现既定项目调度目标的最优化^[1]. RCPSP具有广泛的应用背景, 譬如软件开发、药品研发、地产开发、半导体制造等^[1-5]. 在数学上, RCPSP已被证明为非确定性多项式时间(non-deterministic polynomial-time, NP)难题^[4], 其有效求解算法的设计一直是相关领域的重要研究课题. 尽管目前RCPSP的研究成果已很丰富, 但大多局限于确定性问题. 然而, 在项目的实际执行过程中, 往往由于资源供给不及时、人员办事效率不稳定、机器设备

故障、自然天气变化等因素, 导致项目活动的实际执行时间与预定工期存在偏差. 因此, 不确定RCPSP更符合实际情况, 其研究更具现实意义.

近些年, 项目活动工期为分布已知的随机变量的随机资源约束项目调度问题(stochastic resource-constrained project scheduling problem, SRCPSPP)受到越来越多的关注^[1, 6-8], 但由于问题求解的复杂性, 算法层面的现有研究成果很有限. Stork^[6]提出一种针对随机RCPSP的分枝定界法; Ballestin^[1]提出一种遗传算法(generic algorithm, GA), 并指出以项目中每个活动的期望工期作为活动工期的确定性项目(或称期望项目)的总工期(makespan)与随机环境下的项目makespan期望之间存在偏差, 且偏差与项目活动工期的分

收稿日期: 2014-09-02; 录用日期: 2014-12-26.

[†]通信作者. E-mail: wangling@mail.tsinghua.edu.cn; Tel.: +86 10-62783125-272.

国家重点基础研究发展计划项目(2013CB329503), 国家自然科学基金项目(61174189)资助.

Supported by National Key Basic Research & Development Program of China(2013CB329503) and National Science Foundation of China(61174189).

布的方差正相关; Ballestin和Leus^[7]提出了一种贪婪随机自适应搜索方法, 并探讨了项目总工期的分布与项目活动分布之间的关系; Ashtiani等^[8]提出了一种结合预处理策略的GA, 基于资源冲突对部分项目活动进行优先排序。

作为一种新颖的群体智能优化算法, 果蝇算法(fruit fly optimization algorithm, FOA)^[9]已在金融^[9]、连续优化^[10]、电力负荷预测^[11]、物流服务^[12]、车间调度^[13]、半导体产品最终测试^[14]等问题上得到了成功应用。考虑到果蝇算法原理简单、参数较少, 具有高效求解复杂优化问题的潜力, 本文针对随机RCPSP提出一种基于序的果蝇算法(order-based FOA, OFOA)。考虑到问题的随机因素, 引入序优化(ordinal optimization)思想, 提出基于期望项目的预选机制并采用最优计算量分配技术(optimal computing budget allocation, OCBA)合理评价解。为了均衡全局和局部搜索, 在标准FOA的视觉搜索和嗅觉搜索的基础上增添了协同搜索环节。最后, 通过仿真测试与算法比较验证所提算法的有效性。

2 问题描述(Problem statement)

2.1 随机RCPSP(Stochastic RCPSP)

经典RCPSP可描述如下^[4]: 一个项目包含 $N+2$ 个活动 $J^+ = \{0, 1, \dots, N+1\}$, K 种资源(仅考虑可更新资源), 任意时刻资源 $k(k = 1, 2, \dots, K)$ 的总供给量为 R_k , 活动0和活动 $N+1$ 为虚拟活动, 表示项目的开始和结束, 工期为0, 不消耗资源。项目可用一个非循环的有向图 $G(N, A)$ 表示, 其中: A 为有向弧集, 有向弧 $(i, j)(i, j \in J^+)$ 表示项目活动之间的优先约束关系。项目中活动 j 的工期为 $d_j(j = 1, 2, \dots, N)$, 消耗资源 k 的数量为 $r_{jk}(k = 1, 2, \dots, K)$, 调度目标是合理安排活动的开始时间 $s_j(j \in J^+)$, 使整个项目的最大完工时间(makespan)最小。

随机RCPSP是经典RCPSP的推广, 每个活动 j 的工期是一个分布已知的随机变量 $d_j(j = 1, 2, \dots, N)$ 。由于活动工期不确定, 只有活动完成之后才能确定其工期, 因此活动 j 的开始时间为随机变量 $s_j(j = 1, 2, \dots, N+1)$ 。当所有活动执行完成之后, 得到确定性的执行工期向量

$$\vec{D} = [d_1 \ d_2 \ \dots \ d_N],$$

向量 \vec{D} 为项目的一个具体实现, 称为一个场景(scenario)。随机RCPSP的数学模型可表示如下:

$$\min : E\{s_{N+1}\}, \quad (1)$$

$$\text{s.t. } s_i + d_j \leq s_j, \quad \forall (i, j) \in A, \quad (2)$$

$$\sum_{i \in A_c(t)} r_{ik} \leq R_k, \quad \forall k \in K, \quad (3)$$

其中 $A_c(t)$ 表示 t 时刻执行的活动集合。

2.2 调度策略(scheduling policy)

求解RCPSP时, 搜索操作往往针对项目的活动序列而非实际调度, 因此需要通过解码过程将活动序列转化为可执行的调度, 并进行性能评价。经典RCPSP的求解算法一般采用调度生成机制(schedule generation scheme, SGS)^[4]实现活动序列到可行调度的映射, 通过确定每个活动的开始时间进而得到最终的可行调度。鉴于随机RCPSP的不确定性, 在项目活动完成前难以确定每个活动的工期, 也就不能通过SGS得到活动序列对应的可行调度。因此, 求解随机RCPSP时通常采用调度策略(scheduling policy)^[8]逐步选择活动来获得执行项目的具体调度。采用不同的调度策略可能得到不同的调度方案, 调度的性能也将不同, 即调度策略直接影响求解算法中对解的性能评价。目前已有的调度策略^[8]包括活动策略(activity based policy, 记 C^{AB})、资源策略(resource based policy, 记 C^{RB})、最早开始策略(earliest start policy)、预选策略(preselective policy)、线性预选策略(linear pre-selective policy)、预处理策略(pre-processor policy)。研究表明^[11], C^{AB} 和 C^{RB} 通常优于其他调度策略。就平均性能而言, Ashtiani等^[8]指出 C^{RB} 要优于 C^{AB} , 尤其对活动工期分布的方差较大的问题。因此, 本文采用 C^{RB} 调度策略。

3 果蝇算法(FOA)

果蝇算法^[9]是一种模拟果蝇觅食行为的生物学原理的群体智能搜索算法。算法将搜索寻优过程模拟为果蝇种群的觅食过程, 基于果蝇觅食行为中对嗅觉和视觉的运用, 设计对应的嗅觉和视觉搜索操作, 通过对食物源的迭代搜索, 获得优化问题的满意解。果蝇算法的流程如下^[9]:

步骤 1 初始化种群中心位置。

步骤 2 嗅觉搜索。在种群中心位置附近随机产生 S 个邻域解。

步骤 3 评价个体。通过评价机制计算每个个体的适应度值。

步骤 4 视觉搜索。选择最优邻域解, 更新为种群中心位置。

步骤 5 若终止准则满足, 则输出最优解; 否则, 转至步骤2。

4 随机RCPSP的OFOA(OFOA for stochastic RCPSP)

设计求解随机RCPSP的OFOA的主要思想如下: 第一, 采用活动列表编码表征种群个体; 第二, 采用基于后悔值偏好的随机抽样法和最晚开始时间优先规则^[15]初始化种群; 第三, 针对工期的不确定性, 引入OCBA技术进行合理评价, 并采用基于期望项目的预

选机制粗筛早期解;第四,在嗅觉搜索阶段采用交换操作生成邻域解,在视觉搜索阶段采用保优更新机制更新种群;第五,嵌入协作进化机制增强算法的全局搜索能力。

4.1 编码与评价(Encoding and evaluation)

研究表明^[3],活动列表编码通常优于其他编码方式.随机RCPSP作为经典RCPSP的一种推广,保留了RCPSP的许多特性,因此OFOA采用活动列表编码,即每个个体用活动列表 $\pi = (a_1, \dots, a_N)$ 表示,其中 a_j 表示第 j 个调度的活动.现有随机RCPSP的研究^[7]大多采用活动列表在 NS 个场景 $\vec{D}_j(j = 1, 2, \dots, NS)$ 下的平均总工期作为近似期望工期,如式(4)所示,其中 NS 通常设置为 $10^{[7]}$.为了便于比较,本文采用相同的评价方式.

$$\tilde{C}_{\max}(\pi) = \sum_{j=1}^{NS} s_{N+1}(\vec{D}_j, \pi) / NS. \quad (4)$$

与文献[1, 7-8]一致,对于算法获得的最终解,额外生成1000个场景,相对准确地计算解的期望工期,作为算法的性能.

4.2 种群初始化(Population initialization)

为了增强算法的全局搜索能力,OFOA在迭代过程中采取NP个中心位置,每个中心位置可视为一个个体,进而构成一个种群.参照Hartmann等^[5]的思路,OFOA采用基于后悔值偏好的随机采样方法和最晚完成时间优先规则进行种群的初始化.具体而言,在每个决策时刻,在一个可选活动集(eligible set, SE)中,根据每个活动的选中概率随机选择一个活动开始执行.每个活动 j 被选中的概率 η_j 如下^[5]:

$$\eta_j = (\mu_j + 1) / \sum_{i \in SE} (\mu_i + 1), \quad (5)$$

$$\mu_j = \max_{i \in SE} \text{LFT}_i - \text{LFT}_j, \quad (6)$$

其中:SE表示所有前继活动已被调度的活动集合,LFT _{j} 表示活动 j 的最晚完成时间.

4.3 预选机制(pre-select scheme)

Ballestin指出^[1],为了评价更多的调度,评价个体时生成的场景数应减少.另外,求解随机RCPSP时研究发现在算法初期,许多调度解的性能很差.因此,对所有初期解均生成 NS 个场景进行相对准确的评价将导致计算量过多浪费于劣质解.在比较求解随机RCPSP的不同算法时,生成的总生成调度数(即评价次数)通常是给定并且有限的.为了合理利用有限的总评价次数,提高最终调度解的质量,在此提出一种基于期望项目的预选机制.简单而言,在使用总评价次数的前 $p\%$ 的过程中,每个个体只采用期望项目进行一次评价得到解的性能,而在后期将采用最优计算量分配技术在多个场景下进行性能估计.

4.4 嗅觉和视觉搜索(Smell & vision based search)

果蝇算法是针对连续优化问题提出的,为了使其能求解RCPSP,必须设计特殊的嗅觉和视觉操作.对于RCPSP,无优先约束关系的活动的相对位置对解的性能影响较大^[15],因此采用交换无优先约束关系的两个活动 m, n 的 $\text{swap}(m, n)$ 操作来实现嗅觉搜索.同时,在嗅觉搜索阶段,每个种群中心位置均产生 S 个邻域个体来加强局部搜索.视觉搜索属于趋化性替换,即种群中每个个体将自身位置更新为其最优邻域个体.OFOA嗅觉和视觉搜索过程的伪代码如下:

FOR $i = 1$ to NP

FOR $j = 1$ to S

随机选择无优先约束关系的活动 m, n

$\text{swap}(m, n)$ 生成领域个体 nFf_j

END FOR

评价生成的领域个体,选择最优领域个体

$BnFf_i$

果蝇 Ff_i 更新为 $BnFf_i$

END FOR

4.5 协作机制(Co-evolution scheme)

传统果蝇算法只依赖嗅觉和视觉操作来更新种群,缺少个体间的协同与交互.为了提高算法的全局搜索能力,在此强调不同个体之间的协作搜索,并针对RCPSP设计如下特殊的两点交叉操作加以实现.

对于每一只果蝇 $Ff_i(i = 1, 2, \dots, NP)$,随机选择种群中另一个体 Ff_r ,同时随机产生 $1 \sim N$ 之间的两个整数 r_1, r_2 ,通过两点交叉得到新的子个体 Ff_s .具体而言,位置 $1, \dots, r_1$ 的活动从 Ff_i 继承,位置 $r_1 + 1, \dots, r_2$ 的活动从 Ff_r 继承,其余位置的活动从 Ff_i 继承.即

$$a_j^s = a_j^i, j = 1, 2, \dots, r_1, \quad (7)$$

$$a_j^s = a_k^r, k = \min\{k : a_k^r \in \{a_{r_1}^i, \dots, a_{r_2}^i\}\}, \quad (8)$$

$$a_j^s = a_j^i, j = r_2 + 1, \dots, N, \quad (9)$$

其中 a_j^i, a_k^r, a_j^s 分别表示果蝇 Ff_i, Ff_r, Ff_s 对应的活动列表中位于位置 j 的活动.

4.6 最优计算量分配(OCBA)

OCBA^[16]是不确定环境下智能分配有限计算量的一种有效技术,业已得到许多成功应用^[17-18].不失一般性,考虑如下随机优化问题:

$$\min_{\theta} J(\theta) = E[L(\theta, \xi)], \quad (10)$$

其中: θ 表示解的集合, ξ 表示不确定性, L 表示样本的评价函数.

对于不确定优化问题,通常采用Monte Carlo实验通过多次独立仿真近似估计目标 J ,即 $\tilde{J} = \sum_{j=1}^T L(\theta, \xi) / T$,其中 T 为仿真次数.显然,增加仿真次数 T 可提高目标的评价准确度,但近似估计准确度的收敛速度小

于 $1/\sqrt{T}^{[16]}$. 在给定总计算量 T 下, 为了改善对一组 K 个解中高性能解评价精度, OBCA采用如下步骤^[16]:

步骤1 设置 $g = 0$, 为每个解分配相同的仿真次数 n_0 , 使得 $N_1^g = N_2^g = \dots = N_K^g = n_0$.

步骤2 如果 $\sum_{i=1}^K N_i^g \geq T$, 则结束.

步骤3 增加仿真次数 Δ , 计算新的仿真量分配 $N_i^{g+1}, i = 1, 2, \dots, K$.

步骤4 $j(j = 1, 2, \dots, K)$ 增加额外的仿真次数 $\max(0, N_i^{g+1} - N_i^g)$, 然后令 $g = g + 1$, 返回步骤2.

执行OCBA时, n_0 和 Δ 的大小设置需合理. 若 n_0 太小, 则导致不同解的优劣关系早熟收敛. 若 Δ 太小, 则分配次数过多而导致过程漫长, 反之则劣解浪费计算量. 通常, $n_0 \in [5, 20], \Delta \in [5, K/10]^{[16]}$.

4.7 OFOA流程(Process of OFOA)

基于上述设计, 给出求解随机RCPSP的FOFA流程, 如图1所示. OFOA包括4个主要环节: 基于无优先约束关系的swap操作产生邻域解的嗅觉搜索过程、基于期望项目的预选机制和OCBA的评价过程、基于保优机制的视觉搜索过程、基于两点交叉的协作进化搜索过程.

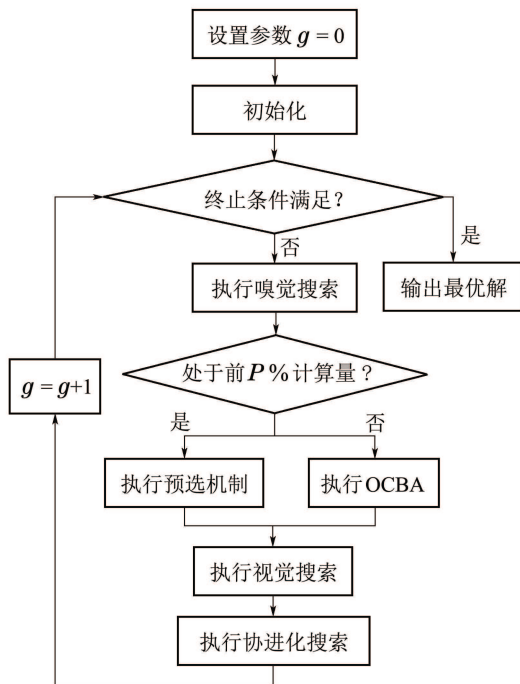


图1 OFOA的流程图

Fig. 1 Flowchart of FOA

5 仿真结果与比较(Simulation results and comparisons)

5.1 实验设置(Experiments setting)

目前学术界关于随机RCPSP的研究很少, 已有的代表性研究均采用PSPLIB^[15]中的J120作为测试集, J120包括60组每组10个共600个测试问题, 每个问题

均包含120个活动, 以确定性工期 d_j^* 作为活动 j 的期望工期, 采用不同分布来测试算法性能. 现有文献考虑如下5种分布, 包括均匀分布

$$U_1 = U(d_j^* - \sqrt{d_j^*}, d_j^* + \sqrt{d_j^*}),$$

方差为 $d_j^*/3$; 均匀分布 $U_2 = U(0, 2d_j^*)$, 方差为 $d_j^{*2}/3$; 指数分布 $\exp(1/d_j^*)$, 方差为 d_j^{*2} ; 区间 $[d_j^*/2, 2d_j^*]$ 内的两种Beta分布:

$$B1 = \text{Beta}\left(\frac{d_j^*}{2} - \frac{1}{3}, d_j^* - \frac{2}{3}\right), B2 = \text{Beta}\left(\frac{1}{6}, \frac{1}{3}\right).$$

显然, U_1 和 B_1 方差较小, U_2 和 B_2 方差水平中等, 而 \exp 方差较大. 在Intel Core2(TM) Q9550 (2.83 GHz) 计算机上采用C++实现OFOA算法, 并采用基于期望项目关键路径长度的平均偏差百分比(APD)来评价算法性能.

$$\text{APD} = \frac{1}{R} \sum_{j=1}^R \frac{E[s_{N+1}(D, \pi)_j] - LB_j}{LB_j}, \quad (11)$$

其中: $R = 600$, 即J120的测试问题数; LB_j 为测试问题 j 的期望项目的关键路径长度^[1,7-8]; 期望工期 $E(s_{N+1})$ 用1000次随机仿真所得工期平均值近似. 为了公平比较算法, 与文献[1,7-8]一致, 分别设置5000和25000次最大调度生成数作为算法的终止条件. 在一个场景下评价一次解, 即为生成一次调度.

5.2 参数设置(Parameters setting)

OFOA有3个关键的参数: 种群规模NP、嗅觉搜索生成邻域解个数 S 、执行预选机制的总评价次数比例 $p\%$. 在此利用试验设计方法(design of experiment, DOE)^[19]探讨参数对算法性能的影响. 首先考虑方差适中的 U_2 分布, 从每组J120问题中随机选取一个问题(共60个问题)开展试验, 各参数均选取4个水平值, 如表1所示, 从而建立规模为 $L_{16}(4^3)$ 的正交试验表. 算法在每组参数组合下均独立运行20次, 以5000次调度为终止条件, 求解60个问题所得平均APD作为评价指标. 正交表和所得评价指标如表2所示. 根据正交表, 计算各参数的极差和重要程度, 如表3所示, 进而绘制各参数对算法性能的影响趋势, 如图2所示. 由表3可见, 参数 S 的极差最大, 这表明嗅觉搜索生成邻域解的数量对算法影响性能最大, 原因在于 S 决定了局部探索的深度, 并间接影响全局搜索的广度. 影响程度其次的是用于执行预选机制的总评价次数比例 $p\%$.

表1 参数水平

Table 1 Parameter values

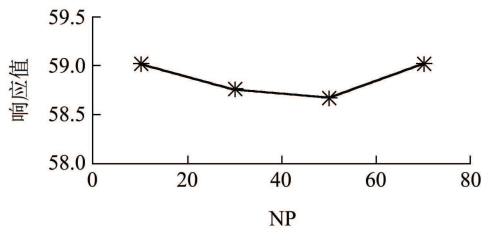
参数	水平			
	1	2	3	4
NP	10	30	50	70
S	1	3	5	7
p	5	10	15	20

表2 正交表和APD
Table 2 Orthogonal array and APD values

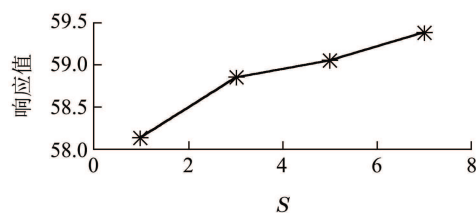
参数组合 编号	水平			APD/%
	NP	S	p	
1	1	1	1	58.65
2	1	2	2	58.27
3	1	3	3	59.11
4	1	4	4	60.02
5	2	1	2	57.69
6	2	2	1	58.43
7	2	3	4	59.55
8	2	4	3	59.33
9	3	1	3	59.75
10	3	2	4	59.54
11	3	3	1	59.12
12	3	4	2	58.26
13	4	1	4	58.49
14	4	2	3	59.18
15	4	3	2	58.42
16	4	4	1	59.97

表3 各参数响应值
Table 3 Response value of parameters

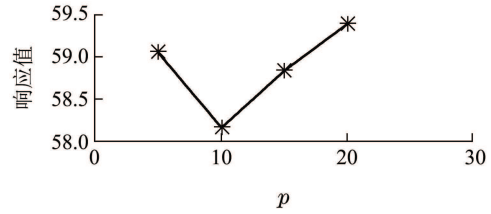
水平	参数		
	NP	S	p
1	59.01	58.15	59.04
2	58.75	58.86	58.16
3	58.67	59.05	58.84
4	59.02	59.39	59.40
极差	0.35	1.25	1.24
等级	3	1	2



(a) NP对算法性能的影响



(b) S对算法性能的影响



(c) p对算法性能的影响

图2 各参数对算法性能影响的水平趋势

Fig. 2 Factor level trend of the parameters

由图2可知,若 p 太小,则评价次数过多浪费于劣质解;反之,则算法早熟收敛,停滞于有限计算量下评价不准确而最终评价性能不好的解.种群规模对算法的性能影响较小,但太小或太大都不利于全局搜索.

进一步,采用表1的参数水平在其他4种分布下进行类似的DOE试验.试验发现,对于不同的分布算法,最佳的参数组合相同,即 $NP = 50, S = 1, p = 10$.因此,后文将基于此参数设置开展进一步的性能测试和算法比较研究.

5.3 实验结果与比较(Results and comparisons)

首先,为了验证预选机制的有效性,比较OFOA与不加入预选机制的FOA(即直接运用OCBA)的性能,结果如表4所示.

表4 OFOA与FOA的比较
Table 4 Comparisons of OFOA with FOA

总调度数	分布	算法				
		U_1	U_2	exp	B_1	B_2
5000	FOA	47.52	58.14	75.35	47.95	58.55
	OFOA	47.18	57.61	74.58	47.77	58.31
25000	FOA	47.16	57.57	74.43	47.21	57.92
	OFOA	47.14	57.57	74.40	47.20	57.92

由表4可见,当总调度数较少时(5000总调度数),加入预选机制能提高算法的性能;当总计算量充足时(25000总调度数),预选机制对于算法性能的提高作用有限.这与设计预选机制的出发点一致,预选机制就是为了节省迭代初期劣质解的评价次数.当计算量限制较严时,预选机制节约的计算量能使算法评价更多的有效解,从而提高算法的最终性能.

其次,为了验证OFOA的有效性,将其与已有求解随机RCPSP的代表性算法进行比较,包括ABGA^[1], ABGR^[7], PPGA^[8].不同总调度数下的统计结果的比较如表5和表6所示,其中NA表示未给出相应结果.

由表5-6可见, OFOA求解随机RCPSP时在总体上表现出较好的性能.5种不同的分布下, OFOA的性能均优于ABGA与PPGA.当活动工期的分布方差中等及以上时,即 U_2, exp 和 B_2 分布, OFOA的性能优于

ABGR. 但方差较小的 U_1 和 B_1 分布, ABGR的性能要优于OFOA. 可见, 采用资源调度策略的OFOA更适合求解活动工期分布方差较大的随机RCPSP, 而当活动工期分布方差较小时, 采用活动调度策略的算法将更有效.

表5 总调度数5000下的算法比较

Table 5 Comparisons with 5000 schedules

分布	算 法				
	U_1	U_2	exp	B_1	B_2
ABGA	51.49	78.65	120.22	—	—
ABGR	46.84	72.58	114.42	47.17	75.97
PPGA	48.86	58.91	76.03	49.01	58.82
OFOA	47.18	57.61	74.58	47.77	58.31

表6 总调度数25000下的算法比较

Table 6 Comparisons with 25000 schedules

分布	算 法				
	U_1	U_2	exp	B_1	B_2
ABGA	49.63	75.38	116.83	—	—
ABGR	45.21	70.95	112.37	45.60	74.17
PPGA	47.21	58.07	74.56	47.25	57.95
OFOA	47.14	57.57	74.40	47.20	57.92

6 结论(Conclusions)

针对随机RCPSP问题本文提出了一种基于序的果蝇算法OFOA, 采用针对RCPSP的离散嗅觉和视觉搜索操作实现果蝇算法, 并增加设计了协作搜索, 同时设计预选机制并采用OCBA技术有效处理随机因素. 仿真结果和算法比较验证了所提预选机制和算法的有效性, 尤其对于分布方差较大的情况. 进一步的工作是: 一方面针对随机RCPSP探讨活动调度策略下算法的性能, 进而提出自适应的调度策略使用机制, 另一方面开展针对随机RCPSP的鲁棒性能指标或多目标下的有效算法设计研究.

参考文献(References):

- [1] BALLESTIN F. When it is worthwhile to work with the stochastic RCPSP? [J]. *Journal of Scheduling*, 2007, 10(3): 153 – 166.
- [2] HERROELEN W, REYCK B D, DEMEULEMEESTER E. Resource-constrained project scheduling: a survey of recent developments [J]. *Computers & Operations Research*, 1998, 25(4): 279 – 302.
- [3] HARTMANN S, KOLISCH R. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem [J]. *European Journal of Operational Research*, 2000, 127(1): 394 – 407.
- [4] BRUCKER P, DREXL A, MOHRING R, et al. Resource-constrained project scheduling: notation, classification, models, and methods [J]. *European Journal of Operational Research*, 1999, 112(1): 3 – 41.
- [5] HARTMANN S, BRISKORN D. A survey of variants and extensions of the resource-constrained project scheduling problem [J]. *European Journal of Operational Research*, 2010, 207(1): 1 – 14.
- [6] STORK F. Branch-and-bound algorithms for stochastic resource-constrained project scheduling [R]. Berlin: Technical University of Berlin, 2000.
- [7] BALLESTIN F, LEUS R. Resource-constrained project scheduling for timely project completion with stochastic activity durations [J]. *Production and Operation Management*, 2009, 18(4): 459 – 474.
- [8] ASHTIANI B, LEUS R, ARYANEZHAD M B. New competitive results for the stochastic resource constrained project scheduling problem: exploring the benefits of pre-processing [J]. *Journal of Scheduling*, 2011, 14(2): 157 – 171.
- [9] PAN W T. A new fruit fly optimization algorithm: taking the financial distress model as an example [J]. *Knowledge-Based Systems*, 2012, 26(2): 69 – 74.
- [10] PAN Q K, SANG H Y, DUAN J H, et al. An improved fruit fly optimization algorithm for continuous function optimization problems [J]. *Knowledge-Based Systems*, 2014, 62(1): 69 – 83.
- [11] LI H, GUO S, LI C, et al. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm [J]. *Knowledge-Based Systems*, 2013, 37(1): 378 – 387.
- [12] LIN S M. Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network [J]. *Neural Computing & Applications*, 2013, 22(3/4): 783 – 791.
- [13] 郑晓龙, 王凌, 王圣尧. 求解置换流水线调度问题的混合离散果蝇算法 [J]. *控制理论与应用*, 2014, 31(2): 159 – 164. (ZHENG Xiaolong, WANG Ling, WANG Shengyao. A hybrid discrete fruit fly optimization algorithm for solving permutation flow-shop scheduling problem [J]. *Control Theory & Applications*, 2014, 31(2): 159 – 164.)
- [14] ZHENG X L, WANG L, WANG S Y. A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem [J]. *Knowledge-Based Systems*, 2014, 57(1): 95 – 103.
- [15] KOLISCH R. Serial and parallel resource-constrained project scheduling methods revisited: theory and computation [J]. *European Journal of Operational Research*, 1996, 90(2): 320 – 330.
- [16] CHEN C H, LIN J, YUCESAN E, et al. Simulation budget allocation for further enhancing the efficiency of ordinal optimization [J]. *Discrete Event Dynamic Systems*, 2000, 10(3): 251 – 270.
- [17] WANG L, ZHANG L, ZHENG D Z. Genetic ordinal optimization for stochastic flow shop scheduling [J]. *International Journal of Advanced Manufacturing Technology*, 2005, 27(1/2): 166 – 173.
- [18] ZHANG R, SONG S, WU C. A two-stage hybrid particle swarm optimization algorithm for the stochastic job shop scheduling problem [J]. *Knowledge-Based Systems*, 2012, 27(1): 393 – 406.
- [19] MONTGOMERY D C. *Design and Analysis of Experiments* [M]. Hoboken: John Wiley and Sons, 2005.

作者简介:

郑晓龙 (1988–), 男, 博士研究生, 主要研究方向为智能优化理论与算法, E-mail: zhengxl11@mails.tsinghua.edu.cn;

王凌 (1972–), 男, 教授, 博士生导师, 主要研究方向为优化和调度理论与方法, E-mail: wangling@mail.tsinghua.edu.cn.