

一种求解 RCPSP 的协同进化分布估计算法

陈 旺, 史彦军, 滕弘飞

(大连理工大学机械工程学院, 辽宁 大连 116023)

摘 要: 针对大规模资源受限项目调度问题计算复杂的特点, 提出一种合作式协同进化分布估计算法(CCEDA)。将合作式协同进化框架与分布估计算法相结合, 将复杂问题分解为子问题, 利用改进的分布估计算法对每个子问题进行协同优化求解。为提高分布估计算法的局部搜索能力, 给出一种对解进行局部搜索的方法。将 CCEDA 用于求解标准问题库 PSPLIB, 并与 GAPS、GA-DBH、GA-hybrid 与 GA-FBI 算法进行比较, 结果证明 CCEDA 拥有更好的求解性能。

关键词: 资源受限项目调度问题; 项目调度; 分解策略; 协同进化; 分布估计算法; 合作式协同进化分布估计算法

Coevolutionary Estimation of Distribution Algorithm for Solving Resource-constrained Project Scheduling Problem

CHEN Wang, SHI Yan-jun, TENG Hong-fei

(School of Mechanical Engineering, Dalian University of Technology, Dalian 116023, China)

【Abstract】 This paper presents Cooperative Coevolutionary Estimation of Distribution Algorithm(CCEDA) to solve Resource-constrained Project Scheduling Problem(RCPSP). It integrates the cooperative co-evolutionary framework and Estimation of Distribution Algorithm(EDA), decomposes RCPSP into several sub-problems, and then applies improved EDA to cooperatively solve these sub-problems. In order to enhance the local search ability of EDA, it gives a local search method for solutions. CCEDA is compared with GAPS, GA-DBH, GA-hybrid and GA-FBI, and experimental results on PSPLIB prove that CCEDA has better performance.

【Key words】 Resource-constrained Project Scheduling Problem(RCPSP); project scheduling; decomposition strategy; coevolutionary; Estimation of Distribution Algorithm(EDA); Cooperative Coevolutionary Estimation of Distribution Algorithm(CCEDA)

DOI: 10.3969/j.issn.1000-3428.2011.14.044

1 概述

资源受限项目调度问题(Resource-constrained Project Scheduling Problem, RCPSP)是运筹学、管理科学、产品设计计划与调度等领域的一个热点问题, 倍受国内外学者的关注。目前求解 RCPSP 问题的算法主要包括三大类^[1]: 精确类算法, 基于优先规则的启发式算法和元启发式算法。其中, 精确类算法在理论上能够得到 RCPSP 问题的最优解, 但只适合求解小规模问题; 基于优先规则的启发式算法虽然易于实现且效率较高, 但解的质量难以保证; 元启发式算法具备鲁棒性强、适合求解大规模问题等优点, 且计算速度快, 可以在计算质量和计算效率上取得很好的平衡, 因此, 在实际求解大规模问题中得到广泛的应用。

分布估计算法(Estimation of Distribution Algorithm, EDA)^[2]作为一种较新的演化算法, 已经成功应用于组合优化问题的求解。EDA 通过建立设计变量的概率模型, 描述设计变量之间的相互关系, 对搜索过程起到宏观的指导作用。但研究表明, EDA 作为一种全局搜索算法, 其局部搜索能力较弱, 收敛速度较慢。并且对于大规模的 RCPSP 问题, 由于 EDA 在算法每次迭代的过程中都需要建立种群个体的概率模型并进行采样, 因此计算量大, 耗时较长。如何有效提高 EDA 求解大规模 RCPSP 问题的效率是关键问题。

协同进化方法^[3]是一种求解大规模问题的有效途径, 该方法遵从“分而治之”的理念, 先按一定方法将复杂问题分解为多个子问题, 然后分别求解, 由此降低问题的复杂度。

因此, 本文针对大规模 RCPSP 问题的特点, 采取协同进化的策略, 将大规模问题分解为多个规模较小的子问题, 然后利用改进的 EDA 对每个子问题进行求解。

2 资源受限项目调度问题

RCPSP 问题可描述为^[1]:

项目包含任务集合 $V = \{j_0, j_1, L, j_{J+1}\}$, 其中, 0 和 $J+1$ 分别代表项目的开始和结束。完成项目需要 K 种可更新资源, 其中, 第 $k(k \in K)$ 种资源在项目执行期间任一时刻的资源总量为 $R_k(k=1, 2, L, K)$ 。对于任务 $j \in V$, 工期为 d_j , 其中, $d_0 = d_{J+1} = 0$, 完成任务 j 所需第 k 种资源的需求量为 $r_{jk}, k=1, 2, L, K$, 其中, $r_{0k} = r_{J+1k} = 0$ 。 d_j 、 r_{jk} 和 R_k 都是正整数。

RCPSP 问题的数学模型可以描述为^[1]:

$$\min F_{J+1} \quad (1)$$

s.t.

$$F_h \leq F_j - d_j, j=1, 2, L, J+1, h \in P_j \quad (2)$$

$$\sum_{j \in A(t)} r_{jk} \leq R_k, k \in K, t \geq 0 \quad (3)$$

$$F_j \geq 0, j=1, 2, L, J+1 \quad (4)$$

基金项目: 国家自然科学基金资助项目(60674078, 50975039)

作者简介: 陈 旺(1983—), 男, 博士研究生, 主研方向: 智能优化算法, 机械产品设计与流程规划; 史彦军, 讲师、博士; 滕弘飞, 教授、博士生导师

收稿日期: 2010-12-29 **E-mail:** dgjx309@163.com

其中, F_j 为任务 j 的完工时间; $A(t) = \{j \in V \mid F_j - d_j \leq t < F_j\}$ 为 t 时刻正在执行任务的集合, 则一个调度 S 的完工时间为 (F_1, F_2, L, F_n) 。式(1)为目标函数, 表示优化目标是项目的总工期最短, 式(2)、式(3)分别表示任务之间的时序约束和资源约束。

3 协同进化分布估计算法

合作式协同进化框架^[3]是一种开放的、有效的、具有一定通用性的计算框架, 为处理大规模和复杂问题提供了一种有效的方法。本文将改进的分布估计算法和合作式协同进化框架相结合, 给出合作式协同进化分布估计算法(Cooperative Coevolutionary Estimation of Distribution Algorithm, CCEDA)用于求解 RCPSP 问题。它将大规模问题分解为多个规模较小的子问题, 每个子问题用一个子种群进行进化, 通过不同种群的个体协同评价, 实现大规模复杂问题的优化求解。使用 CCEDA 求解 RCPSP 问题有以下 3 个问题需要解决: (1)问题的分解形式; (2)子种群个体适应度的评价方法; (3)子种群的进化机制。

3.1 问题的分解

利用协同进化方法求解复杂大规模问题需要解决的一个问题是如何将大规模问题分解为若干子问题, 并使这些子问题之间的耦合程度最低。对于 RCPSP 问题来说, 由于资源许用量的限制, 并行任务之间存在对有限资源的竞争关系, 如何合理安排项目各阶段并行任务的开始时间是问题求解的关键。一个最直接有效的分解方法是将项目各阶段的并行任务聚合为一个子问题。

设原问题包含任务数为 n , 分解后的子问题数目为 S , 每个子问题包含的任务集合为 $A_s (s=1, 2, L, S)$ 。问题分解步骤如下:

(1)不考虑资源约束, 使用关键路径法对项目进行调度。设获得的调度方案的工期为 T , 每个任务的开始时间为 $st_i (i=1, 2, L, n)$, 结束时间为 $ft_i (i=1, 2, L, n)$ 。

(2)将整个项目按照时间平均分为 S 个阶段, 第 $s (s=1, 2, L, S)$ 个阶段的时间区间为 $[(s-1) \cdot (T/S), (s \cdot T)/S]$ 。

(3)根据每个任务所在阶段, 即任务的中间时间 $\frac{st_i + ft_i}{2}$ 所在阶段, 将任务添加到该阶段对应子问题的任务集合 $A_s (s=1, 2, L, S)$ 中。

以图 1 所示的项目为例, 利用关键路径法得到的调度方案如图 2 所示, 其中, $stage_1$ 和 $stage_3$ 阶段内阴影部分的任务分别表示第 1 个阶段和第 3 个阶段的任务集合; 其余任务为第 2 个阶段的任务集合。

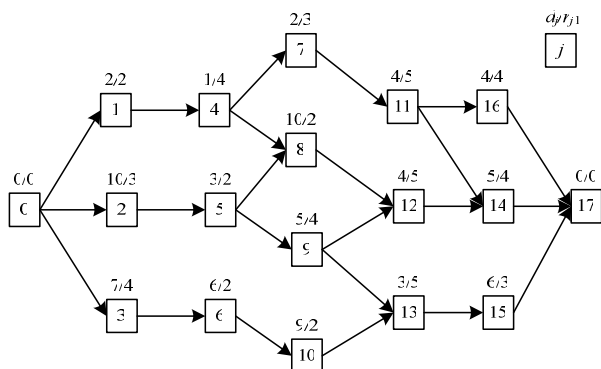


图1 项目网络图

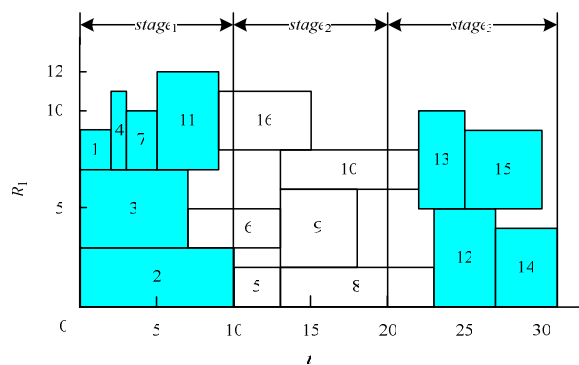


图2 利用关键路径法得到的调度方案

假设分解后子问题数目为 3, 首先将项目分为 3 个阶段 $stage_k (k=1, 2, 3)$, 则由图 2 可知, 利用本文方法将原问题分解后, 每个子问题对应的任务集合分别为:

$$A_1 = \{1, 2, 3, 4, 7, 11\}$$

$$A_2 = \{5, 6, 8, 9, 10, 16\}$$

$$A_3 = \{12, 13, 14, 15\}$$

3.2 子种群个体适应度的评价

对于子种群中个体进行评价时, 需要从其他子种群中选择合作个体组成完整解进行评价。随着 RCPSP 问题规模的增大, 适应度函数计算非常耗时, 而在工程实践中往往需要在有限时间内获得满意解, 因此, 本文采用最优选择法, 在对当前子种群个体进行评价时, 从其他子种群中选择适应度最高的个体作为合作个体, 与当前个体组成完整解进行评价, 并将该完整解的适应度函数值作为当前子种群个体的适应度函数值。

需要说明的是, 在种群初始化阶段, 由于子种群中个体的适应值尚未确定, 因此无法采用最优选择法。本文在种群初始化时采用随机选择法。

3.3 子种群的进化机制——改进的分布估计算法

本文采用改进的分布估计算法对分解后得到的每个子问题进行求解。EDA 是在演化过程中通过统计学习的方式建立种群个体的概率模型, 然后根据建立的概率模型在解空间中采样获得新的种群, 如此反复迭代, 实现种群的进化。EDA 主要包含 3 个步骤:

- (1)选择样本解。
- (2)根据选择的样本解构建分布概率模型。
- (3)根据构建的概率模型进行采样, 得到新解。

另外, 本文针对 EDA 局部搜索能力弱的缺点, 利用单纯形法对迭代过程中获得的迭代最优解进行局部搜索, 以更有效地求解 RCPSP 问题。

3.3.1 解的编码和解码

CCEDA 算法采用随机键编码, 子问题中的每个解都可以表示为一个随机数序列:

$$X_s = \langle r_1, r_2, L, r_{s_n} \rangle, s=1, 2, L, S$$

其中, $0 < r_j < 1 (j=1, 2, L, s_n)$, s_n 为子问题 s 包含的任务数。

对于随机键编码的解, 需要采用调度产生方案对其进行解码。有 2 种调度产生方案: 串行调度产生方案和并行调度产生方案^[1]。考虑到这 2 种调度方案各具优缺点, 本文采用如下策略: 在每次解码时随机生成一个数 $r_{SGS} (0 \leq r_{SGS} \leq 1)$, 如果 $r_{SGS} \leq 0.5$, 选择串行调度产生方案, 如果 $r_{SGS} > 0.5$, 选择并行调度产生方案。

3.3.2 样本解选择方法

本文采用2-锦标赛选择法从种群中选择代表解,为EDA构建概率模型提供样本解。在2-锦标赛选择法中,每次随机地从种群中挑选2个个体,然后根据适应值将较好的个体选作样本解。重复这个过程多次,直到样本解的个数满足要求,本文取样本解个数为种群个数的一半,设种群大小为 NP ,则样本解数量为 $NP/2$ 。

3.3.3 概率模型的构建方法

从种群中挑选出样本解后,需要根据问题特点,构建适合问题的概率模型。因为在很多现实情况下,变量的分布都是高斯分布或近似高斯分布,所以本文也假设设计变量是基于高斯分布的,采用带对角协方差矩阵的高斯模型(Gaussian Model with Diagonal Covariance Matrix, GM/DCM)^[2]建立概率模型。GM/DCM综合考虑了各随机变量间的相互联系,以矩阵中的每一个元素来表示随机向量不同分量之间的协方差,可以更好地描述不同设计变量之间的相关性,并利用该相关性指导搜索过程。

3.3.4 采样算法

在EDA算法中,新个体的产生是通过对描述概率模型的联合分布函数进行采样而实现的。本文采用的GM/DCM其每一维设计变量都服从 $N(\mu_j, \sigma_j)$ 描述的正态分布。本文采用Box-Muller公式^[2]作为采样方法。

3.3.5 局部搜索方法

针对EDA算法局部搜索能力弱的缺点,本文利用局部搜索能力较强的单纯形法^[4]对每次迭代得到的当前最优解的邻域进行局部搜索。单纯形法无需函数的梯度信息,是解决非线性无约束优化问题的有效方法。

4 PSPLIB 问题库验证实验

本文参考研究中最常用的验证方式,采用标准问题库PSPLIB来验证CCEDA算法的有效性。算法采用Java编写,编译器为JDK 1.6,运行环境为:PC一台,CPU为奔腾双核1.86 GHz,内存为2 GB,操作系统为Windows XP。

4.1 标准问题库 PSPLIB

本文采用使用最广泛的PSPLIB问题库中的J60和J120 2个问题集对CCEDA算法求解RCPSP问题的有效性进行验证。目前这2个问题集还没有获得全部问题实例的最优解,而只给出了下界。

4.2 实验设置

在前期实验中,采用不断实验的方法来确定CCEDA算法参数的最优设置,种群大小 $NP=100$,子问题数目 $S=4$ 。为了与其他算法公平比较,采用与其他算法相同的算法结束准则,即将算法迭代过程中产生解的最大个数限制为1 000、5 000和50 000。

4.3 实验结果与分析

CCEDA与当前求解效果较好的算法的计算结果比较如表1、表2所示,其他算法的计算结果来自文献[1]。其中,表1比较了CCEDA与其他算法求解J60问题集时获得的最好结果与最优解或下界的平均偏差;表2为各算法求解J120问题集的实验结果。由表1可以看出,当算法生成个体总数为1 000时,CCEDA获得最好结果的平均偏差为11.74%,

仅排名第5;当生成个体总数为5 000时,CCEDA获得的最好结果的平均偏差为11.03%,排名第2;当生成个体总数为50 000时,平均偏差为10.64%,在所有算法中排名第1。由表2可以看出,当算法生成个体总数分别为1 000、5 000和50 000时,CCEDA获得的最好结果的平均偏差分别为33.59%、32.23%和30.65%,排名都为第1。由以上结果可知,对于大规模(任务数为120)或中等规模(任务数为60)的RCPSP问题而言,CCEDA可以获得比其他算法更好的计算结果,表明CCEDA适合求解大规模RCPSP问题。

表1 求解J60问题集所得结果与最优解或下界的平均偏差(%)

个体总数	CCEDA	GAPS	GA-DBH	GA-hybrid	GA-FBI
1 000	11.74	11.72	11.45	11.56	12.21
5 000	11.03	11.04	10.95	11.10	11.27
50 000	10.64	10.67	10.68	10.73	10.74

表2 求解J120问题集所得结果与最优解或下界的平均偏差(%)

个体总数	CCEDA	GAPS	GA-DBH	GA-hybrid	GA-FBI
1 000	33.59	34.19	34.07	35.87	35.39
5 000	32.23	32.34	32.54	33.03	33.24
50 000	30.65	30.82	31.24	31.44	31.58

CCEDA算法能取得较好的计算结果主要有3个原因:

(1)采取分解和协同进化的策略,将大规模复杂问题分解为若干子问题,然后利用改进的EDA进行求解,缩小了搜索空间,降低了问题求解难度。

(2)EDA通过分析问题解空间的全局信息和设计变量间的内在联系,对最优解的分布进行预测,从而指导CCEDA算法的全局搜索过程,有助于算法跳出局部最优。

(3)针对EDA局部搜索能力弱的缺点,利用单纯形法对迭代最优解的邻域进行局部搜索,加快了算法收敛速度。

5 结束语

本文给出了一种用于求解经典RCPSP问题的分布估计协同进化算法。该算法采取协同进化的策略,将合作式协同进化框架和分布估计算法相结合,将大规模复杂问题分解为若干子问题,然后利用改进的分布估计算法对子问题进行求解。最后利用标准问题库PSPLIB的单模式问题集进行实验,结果验证了CCEDA算法的有效性。

参考文献

- [1] Kolisch R, Hartmann S. Experimental Investigation of Heuristics for Resource-constrained Project Scheduling: An Update[J]. European Journal of Operational Research, 2006, 174(1): 23-37.
- [2] Larranaga P, Lozano J. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation[M]. [S. l.]: Kluwer Academic Public, 2002.
- [3] 周铁军, 李 阳. Multi-Agent 协同进化算法研究[J]. 计算机工程, 2009, 35(13): 205-207.
- [4] 唐焕文, 秦学志. 实用最优化方法[M]. 大连: 大连理工大学出版社, 2000.

编辑 张 帆