文章编号:1001-9081(2012)08-2165-03

doi:10.3724/SP.J.1087.2012.02165

改进自适应差分进化算法求解大规模整数任务分配

王永皎*

(河南城建学院 计算机科学与工程系,河南 平顶山 467007)

(*通信作者电子邮箱 wangyj111@sohu.com)

摘　要: 针对0-1任务规划模型存在维数灾难的问题,提出一种基于改进自适应差分进化(SADE)算法的大规模整数任务分配算法。首先,将任务分配的0-1规划模型转化整数规划模型,不仅大幅减少了优化变量的维数,还减少了整式约束条件;然后,将常用的变异算子DE/rand/1/bin和DE/best/2/bin结合起来组成新的自适应变异算子,使得自适应差分进化算法既有较快的收敛速度,又降低了变异算子对具体问题的依赖;并用改进自适应差分进化算法求解整数规划。最后,通过典型的任务分配实例验证了算法在优化大规模任务分配的有效性和快速性。

关键词: 自适应差分进化算法;任务分配;0-1规划;整数规划;变异

中图分类号: TP18　　**文献标志码:** A

Improved self-adaptive differential evolution algorithm for large-scale integer task assignment

WANG Yong-jiao*

(*Department of Computer Science and Engineering, Henan University of Urban Construction, Pingdingshan Henan 467044, China*)

Abstract: In order to solve the problem that the general 0-1 task assignment has dimension disaster problem, an integer task assignment based on improved Self-Adaptive Differential Evolution (SADE) algorithm was proposed. Firstly, 0-1 task assignment model was transferred into integer task assignment model, which not only decreased the dimension of variable, but also decreased equation constraints. Then, classical DE/rand/1/bin and DE/best/2/bin mutation operators were added with linear weight, which made the SADE algorithm not only converge quickly, but also decrease independence on concrete problem, and the integer task assignment model was optimized by the improved SADE algorithm. At last, several classic task assignment problems were tested. The experimental results show that the proposed algorithm is effective and speedy on the large-scale task assignment.

Key words: Self-Adaptive Differential Evolution (SADE) algorithm; task assignment; 0-1 programming; integer programming; variation

0 引言

任务分配问题(Task Assignment Problem, TAP)由Stone于1977年首先提出,它涉及在分散系统中将多个任务分配给多个处理器,其目标是将处理器之间的通信成本和工作处理成本的总和降到最低。目前,任务分配模型均是0-1规划模型[1-7],是一个NP难题,在优化大规模的任务分配时,性能欠佳[2]。经过研究发现,任务分配的决策矩阵是一个稀疏矩阵,矩阵的每一行有且仅有一个元素为1,其余元素均为0,因此,本文将每一行中为1的元素在每一行的位置作为优化变量,将任务分配的0-1模型转化为整数模型,可大幅降低优化变量的维数,还放宽了整式约束条件。

近年来,学者们已经提出了许多智能算法来解决TAP,如遗传算法[1]、粒子群算法[2-3]、调和搜寻算法[4]和差分进化算法(Differential Evolution, DE)[5-6]等。DE因原理简单、受控参数少、鲁棒性强等特点,引起越来越多的学者关注。近年来,DE在约束优化计算、聚类优化计算、非线性优化控制、神经网络优化、滤波器设计及其他方面得到广泛应用[7-8]。由于基本的DE算法的参数设置依赖于具体的待优化的问题,并且调整参数是一个很艰难的过程,因此,许多学者研究了参数的自适应调整的问题,即自适应差分进化(Self-Adaptive Differential Evolution, SADE)算法[9-13]。文献[14]指出SADE算法的性能与变异算子的选择有关,在优化某一类问题时,一种变异算子性能较好,在优化另一类问题时,另一种变异算子性能较好。因此,SADE算法只解决了参数的自适就调整,并未解决变异算子的自适应选择的问题。本文研究将DE/rand/1/bin和DE/best/2/bin结合起来组成新的变异算子,在前期,DE/rand/1/bin起主要作用以保持种群的多样性;在后期,DE/best/2/bin起主要作用以加快算法的收敛,并将改进的SADE用于求解任务分配问题。

1 工作分配问题的整数模型

工作分配的数学模型有多种方式,本文引用文献[2]的模型:

$$\min Q(x) = \sum_{i=1}^{r}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{r-1}\sum_{j=i+1}^{r} c_{ij}\left(1 - \sum_{k=1}^{n} x_{ik}x_{jk}\right) \tag{1}$$

$$\text{s.t.}\quad \sum_{k=1}^{n} x_{ik} = 1$$

$$\sum_{i=1}^{r} m_i \cdot x_{ik} \leqslant M_k$$

收稿日期: 2012-02-01;**修回日期:** 2012-03-27。

作者简介: 王永皎(1977-),女,河南新乡人,副教授,博士,主要研究方向:人工智能、图像处理。

$$\sum_{i=1}^{r} p_i \cdot x_{ik} \leqslant P_k$$
$$x_{ik} \in \{0,1\}, \forall i,k$$

其中：x_{ik} 是决策变量，如果任务 i 被分配给处理器 k，则 $x_{ik} = 1$，否则，$x_{ik} = 0$；n 是处理器的数目；r 是任务的数目；e_{ik} 是任务 i 在处理器 k 上的执行成本；c_{ij} 是任务 i 和工作 j 在不同处理器上执行时的通信成本；m_i 是任务 i 对它的执行处理器的存储要求；M_k 是处理器 k 的存储能力；p_i 是任务 i 对它的执行处理器的处理要求；P_k 是处理器 k 的处理能力。

由 $\sum_{k=1}^{n} x_{ik} = 1$ 可知，决策矩阵 $\boldsymbol{X} = [x_{ik}]_{r \times n}$ 中的每一行都有且仅有一个元素为 1，其余元素均为 0。设 $\boldsymbol{L} = [l_1, l_2, \cdots, l_r]$，$l_i \in [1,n]$，$l_i$ 表示第 i 行中的第 l_i 列的元素为 1，然后，通过 $\boldsymbol{L}$ 可以构造出决策矩阵 $\boldsymbol{X}$，计算目标函数值和判断约束条件是否满足。$\boldsymbol{X}$ 中共有 $r \times n$ 个 0-1 变量，而向量 $\boldsymbol{L}$ 只有 r 个整数变量，仅为 $\boldsymbol{X}$ 的 $1/n$，同时，整式约束条件已经满足，使 TAP 大大简化。在计算目标函数值和不等式约束条件时，依然采用决策矩阵 $\boldsymbol{X}$。对约束的处理采取文献[15]的方法。

2 差分进化算法

算法首先在问题的可行解空间随机初始化种群 $\boldsymbol{X}^0 = (\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \cdots, \boldsymbol{x}_{NP}^0)^{\mathrm{T}}$，$NP$ 为种群规模，$\boldsymbol{x}_i^0 = [x_{i,1}^0, x_{i,2}^0, \cdots, x_{i,D}^0]$ 用于表征问题解，D 为问题的维数。算法的基本思想是：对当前种群进行变异和交叉操作，产生另一个新种群；然后利用基于贪婪思想的选择操作对这两个种群进行一对一选择，从而产生最终的新一代种群。

2.1 变异算子

DE 对每一个在 t 时刻的个体 $\boldsymbol{x}_i^t$ 实施变异得到与其相对应的变异个体 $\boldsymbol{v}_i^{t+1}$，常用的变异算子有：

DE/rand/1/bin 变异算子：

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{x}_{r1}^t + F(\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t) \tag{2}$$

DE/best/2/bin 变异算子：

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{x}_{\mathrm{gbest}}^t + F(\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t + \boldsymbol{x}_{r4}^t - \boldsymbol{x}_{r5}^t) \tag{3}$$

其中：$r1, r2, r3, r4, r5 \in \{1,2,\cdots,NP\}$ 互不相同且与 i 不同；$\boldsymbol{x}_{r1}^t$ 为父代基向量；$(\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t)$ 和 $(\boldsymbol{x}_{r4}^t - \boldsymbol{x}_{r5}^t)$ 称作父代差量；F 为缩放比例因子，$\boldsymbol{x}_{\mathrm{gbest}}^t$ 为种群中适应值最好的个体。DE/rand/1/bin（式(2)）的变异有利于保持种群的多样性，因而全局搜索能力强，但收敛速度慢；DE/best/2/bin（式(3)）的变异局部搜索能力强，精度高，收敛速度快，但会加大算法陷入局部最优点的可能性。

结合这两种不同变异方式的特点，本文提出了一种新的变异方案，其变异操作方程为：

$$\boldsymbol{v}_i^{t+1} = (1-\lambda)(\boldsymbol{x}_{\mathrm{gbest}}^t + F(\boldsymbol{x}_{r4}^t - \boldsymbol{x}_{r5}^t)) + F(\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t) + \lambda \boldsymbol{x}_{r1}^t \tag{4}$$

其中：$\lambda \in [0,1]$，若 $\lambda = 1$，则式(4)退化为式(2)，变成 DE/rand/1/bin；若 $\lambda = 0$，则式(4)退化为式(3)，变成 DE/best/2/bin。将 λ 设置为线性因子，如式(5)所示。λ 在搜索过程中由 1 逐渐变化为 0，使得 DE/rand/1/bin 的权重逐渐减小而 DE/best/2/bin 的权重逐渐增加，在初始阶段有较强的全局搜索能力，尽可能发现多的可能全局最优点；而在后阶段则应有较强的局部搜索能力，以提高算法的精度和收敛速率。从而保证算法既有较强的全局搜索又有较快的收敛速率和搜索精度。

$$\lambda = (T-t)/T \tag{5}$$

其中：T 表示最大迭代次数，t 表示当前迭代次数。

2.2 交叉操作

变异操作产生的个体 $\boldsymbol{v}_i^{t+1}$ 与父代个体 $\boldsymbol{x}_i^t$ 进行交叉操作，以生成个体 $\boldsymbol{u}_i^{t+1}$，即

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & rand(j) \leqslant CR \text{ 或 } j = rnbr(i) \\ x_{i,j}^t, & \text{其他} \end{cases} \tag{6}$$

其中：$rand(j)$ 为 $[0,1]$ 内的均匀分布随机数，CR 为范围在 $[0,1]$ 内的交叉概率；$rnbr(i)$ 为 $\{1,2,\cdots,D\}$ 之间的随机量。

2.3 选择操作

DE 采用“贪婪”的搜索策略，经过变异与交叉操作后生成的实验个体 $\boldsymbol{u}_i^{t+1}$ 与 $\boldsymbol{x}_i^t$ 进行竞争，只当 $\boldsymbol{u}_i^{t+1}$ 的适应度值较 $\boldsymbol{x}_i^t$ 更优时才被选作子代，否则直接将 $\boldsymbol{x}_i^t$ 作为子代，即

$$\boldsymbol{x}_i^{t+1} = \begin{cases} \boldsymbol{u}_i^{t+1}, & f(\boldsymbol{u}_i^{t+1}) < f(\boldsymbol{x}_i^t) \\ \boldsymbol{x}_i^t, & \text{其他} \end{cases} \tag{7}$$

其中：f 为目标函数。

2.4 参数的自适应策略

DE 算法的参数严重依赖于具体的问题，因此，文献[4]研究了参数的自适应策略，这种策略自适应地控制缩放因子和交叉概率，二者的调节方式如下式所示：

$$F_i^{t+1} = \begin{cases} F_1 + rnd_1 \cdot F_{\mathrm{u}}, & rnd_2 < 0.1 \\ F_i^t, & \text{其他} \end{cases} \tag{8}$$

$$CR_i^{t+1} = \begin{cases} rnd_3, & rnd_4 < 0.1 \\ CR_i^t, & \text{其他} \end{cases} \tag{9}$$

其中：F_1 和 F_{u} 表示缩放因子的下界和上界，rnd_1, rnd_2, rnd_3 和 rnd_4 为 $[0,1]$ 内均匀分布的随机数。

3 实验结果及分析

TAP 中，不同工作之间的通信由任务交互表（Task Interactive Graph, TIG）来说明。集合 $G(V,E)$ 用来详细解释 TIG，其中，V 是 r 个节点的集合，用来代表 r 个被执行的任务；每个边缘 $(i,j) \in E$ 与 c_{ij} 有关，它表示仅当工作 i 和 j 被分配给不同处理器时所蒙受的通信成本。Yin 等[2]定义了集合 $G(V,E)$ 的工作交互密度 d，用它来计算 TIG 中工作通信的比例。为了测试算法对不同规模的问题的适应性，Yin 等将 (r, n) 分别设置为(5,3)，(10,6)，(15,9)，(20,12)和(25,15)，并设置了 3 种不同的工作交互密度 d，分别为 0.3，0.5 和 0.8。TAP 的其他参数都是在一定范围内随机产生的，具体可参见文献[1]。

计算机的配置：操作系统为 Windows XP，CPU 为 mobile AMD Sempron Processor 2600+，内存为 DDR 768 MB，仿真环境为 Matlab 7.0。DE 的参数设置为：种群规模 $NP = 60$，交叉概率 $CR = 0.6$，缩放因子 $F_1 = 0.4$，$F_{\mathrm{u}} = 0.7$，0-1 规划模型的最大迭代代数为 2 000，4 000，6 000，8 000 和 8 000（相对于 $(r,$

n)分别设置为(5,3),(10,6),(15, 9),(20,12)和(25, 15)),整数规划模型的最大迭代代数为 200,400,600,800 和 800。本文采用同样的方法产生 15 个问题,每一个问题分别用改进的自适应差分进化算法对 0-1 规划模型和整数规划模型进行优化,并且独立运行 10 次,取平均最优值、方差和平均耗时进行比较,如表 1 所示。

表 1　改进的自适应差分进化算法求解两种任务规划模型的比较

r	n	d	0-1 规划模型				整数规划模型			
			最优值	平均值	最差值	方差	最优值	平均值	最差值	方差
5	3	0.3	247.4	247.4	247.4	0.0	247.4	247.4	247.4	0.0
		0.5	284.9	284.9	284.9	0.0	284.9	284.9	284.9	0.0
		0.8	284.0	284.6	286.0	1.0	284.0	284.0	284.0	0.0
10	6	0.3	565.5	609.2	705.0	45.4	565.5	593.5	646.8	34.2
		0.5	821.9	827.7	841.8	8.2	821.9	821.9	821.9	0.0
		0.8	797.4	833.5	944.6	44.2	797.4	824.2	856.7	17.0
15	9	0.3	904.9	1224.6	1386.6	152.2	763.2	772.6	830.6	21.1
		0.5	2055.2	2501.2	3340.8	443.9	1450.2	1493.4	1513.5	22.9
		0.8	1635.9	1686.2	1778.2	61.4	1588.8	1651.1	1706.5	48.4
20	12	0.3	1652.2	1693.8	1751.3	33.7	1612.2	1723.6	1859.8	71.6
		0.5	1884.0	2165.2	2336.9	156.4	1847.6	1960.8	2195.5	105.5
		0.8	4307.0	4488.4	4780.3	178.2	3171.3	3410.4	3570.6	138.9
25	15	0.3	3466.0	3124.4	3813.2	203.1	2538.6	2396.5	2729.8	110.5
		0.5	4009.0	3466.0	4493.9	312.8	3283.9	3115.1	3385.1	103.8
		0.8	6230.1	5852.1	6817.1	317.9	5612.0	5470.9	5801.2	88.7

从表 1 可以看出,在优化小规模的任务分配问题时,整数规划模型和 0-1 规划模型均能找到最优解,只是整数规划模型的方差小很多;但是随着任务分配的规模的增长,整数规划模型的最优值、平均值、最差值和方差优于 0-1 规划模型的结果,并且任务规模越大,整数规划模型的优势越明显。其主要原因是整数规划模型优化变量的维数只有 0-1 规划模型的 $1/n$,同时满足等式约束条件,大大简化了优化的过程。充分说明将任务分配的 0-1 规划模型转化为整数规划模型不仅可以简化计算,而且还可以加快算法的收敛,这种优势在大规模任务分配时,表现尤为突出。

4　结语

本文将 TAP 的 0-1 模型转化为整数模型,使得 TAP 的优化变量维数不会随任务数和处理器数的增加而呈几何级数增长,仅为 0-1 模型的 $1/n$(n 为处理器数),同时,自然满足等式约束条件;将 DE 的两种常用的变异算子 DE/rand/1/bin 和 DE/best/2/bin 结合起来组成新的变异算子,解决变异算子的自适应选择的问题。本文算法在求解大规模任务分配时,性能尤为突出,具有广泛的应用前景。

参考文献:

[1] HADJ-ALOUANE A B, BEAN J C, MURTY K G. A hybrid genetic/optimization algorithm for a task allocation problem [J]. Journal of Scheduling, 1999, 2(4): 189 - 201.

[2] YIN P-Y, YU S-S, WANG P-P, *et al.* A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems [J]. Computer Standards & Interfaces, 2006, 28(4): 441 - 450.

[3] ZOU DEXUAN, GAO LIQUN, LI S, *et al.* A novel global harmony search algorithm for task assignment problem [J]. Journal of Systems and Software, 2010, 83(10): 1678 - 1688.

[4] ZOU DEXUAN, LIU HAIKUAN, GAO LIQUN, *et al.* An improved differential evolution algorithm for the task assignment problem [J]. Engineering Applications of Artificial Intelligence, 2011, 24(4): 616 - 624.

[5] ZHANG DANDAN, XIE GUANGMING, YU JUNZHI, *et al.* Adaptive task assignment for multiple mobile robots via swarm intelligence approach [J]. Robotics and Autonomous Systems, 2007, 55(7): 572 - 588.

[6] 吴沛锋, 高立群, 邹德旋. 改进的差分进化算法在工作分配中的应用[J]. 东北大学学报: 自然科学版, 2010, 31(12): 1697 - 1700.

[7] 李炜, 张伟. 基于粒子群算法的多无人机任务分配方法[J]. 控制与决策, 2010, 25(9): 1359 - 1364.

[8] AL-ANZI F S, ALLAHVERDI A. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times [J]. European Journal of Operational Research, 2007, 182(1): 80 - 94.

[9] PAN QUAN-KE, SUGANTHAN P N, WANG LING, *et al.* A differential evolution algorithm with self-adapting strategy and control parameters [J]. Computers & Operations Research, 2011, 38(1): 394 - 408.

[10] GOUDOS S K, BALTZIS K B, ANTONIADIS K, *et al.* A comparative study of common and self-adaptive differential evolution strategies on numerical benchmark problems [J]. Procedia Computer Science, 2011, 3: 83 - 88.

[11] 杨启文, 蔡亮, 薛云灿. 差分进化算法综述[J]. 模式识别与人工智能, 2008, 21(4): 506 - 513.

[12] 姚峰, 杨卫东, 张明, 等. 改进自适应变空间差分进化算法[J]. 控制理论与应用, 2010, 27(1): 32 - 38.

[13] 吴亮红, 王耀南, 袁小芳, 等. 自适应二次变异差分进化算法[J]. 控制与决策, 2001, 21(8): 898 - 902.

[14] 辛斌, 陈杰, 彭志红, 等. 基于互补变异算子的自适应差分进化算法[J]. 东南大学学报: 自然科学版, 2009, 39(Supl): 10 - 15.

[15] 吴亮红, 王耀南, 周少武, 等. 采用非固定多段映射罚函数的非线性约束优化差分进化算法[J]. 系统工程理论与实践, 2007, 27(3): 128 - 133.