

基于改进遗传算法的工作流调度研究

王晓军,熊 潇

(南京邮电大学 信息网络研究所,江苏 南京 210003)

摘 要:在资源受限工作流系统中,任务的执行顺序和资源分配对工作流执行时间有很大影响。文中就此问题提出了一种新的方法,把工作流调度分为计划和执行两个阶段,先运用改进的遗传算法对工作流系统中的任务执行顺序和资源分配做好全局优化,然后再按照计划执行,达到执行时间最短的目的。实验结果表明,与动态工作流调度方法以及标准遗传算法比较,在相同工作流中,当存在并发执行的任务时,基于改进后的遗传算法的调度方法能够做到全局资源分配最优,使得整个工作流系统在执行时间方面最短。

关键词:工作流调度;资源分配;遗传算法;精英保留;动态变异

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2013)07-0108-04

doi:10.3969/j.issn.1673-629X.2013.07.027

Research of Workflow Scheduling Based on Improved Genetic Algorithm

WANG Xiao-jun, XIONG Xiao

(Institute of Information Network Technology, Nanjing University of Posts and
Telecommunications, Nanjing 210003, China)

Abstract: In the workflow system of limited resources, task execution sequence and resource allocation has a great effect on execute time of workflow. Present a new method, workflow scheduling is divided into two parts are plan and execute. First, use improved genetic algorithm in workflow scheduling to optimize task execution sequence and resource allocation globally, and then perform as planned to achieve the purpose of shortest execution time. Finally, compare improved genetic algorithm with dynamic workflow scheduling method and canonical genetic algorithm in the same workflow. The results show that when exist concurrent tasks, scheduling method based on improved genetic algorithm can optimize resource allocation globally and make the entire workflow system achieve execution time shortest.

Key words: workflow scheduling; resource allocation; genetic algorithm; elitist strategy; dynamic mutation

0 引言

在工作流管理中资源是一个基础的概念。资源(人或者机器)支持每个工作流任务的执行^[1],在资源不充足时会限制工作流任务的执行,如何找到最合适的资源去执行任务是一个工作流管理系统中最有意义的功能(WFMS)^[2]之一。在工作流系统中,合适的资源管理应该是给每个原子任务匹配最合适的资源。资源分配原则需要做出两个决策:第一,任务的执行顺序;第二,在可用资源中挑选合适的资源,并分配给这些需要执行的任务^[3]。这两个决策使得资源分配问题和工作流调度紧密联系在一起。

在众多工作流管理系统中,任务是在运行期间一个接一个调度的,这些工作流管理系统中采用的动态

方法^[4,5]往往只是对单个任务的执行时间进行优化,并没有把整个工作流系统当做一个整体对其执行完所有任务所需的时间进行优化。文献[6]和文献[7]提出了基于遗传算法的工作流调度算法,但没有考虑到任务之间可能存在的并发关系,适用范围有限。文中提出了一种改进的遗传算法(IGA),用于在工作流执行之前计划好执行时间最短的任务序列和资源组合,达到全局优化的目的,并且在染色体编码上考虑到了任务之间的顺序和并发关系。

1 问题描述

根据文献[3]提出的合适的资源分配原则需要对任务执行顺序和资源如何分配做出决策,在文中这两

收稿日期:2012-09-15

修回日期:2012-12-18

网络出版时间:2013-03-05

基金项目:国家科技支撑计划(2007BAH17B04)

作者简介:王晓军(1968-),女,研究生导师,副研究员,研究方向为分布式计算与应用、云计算、数据库技术等;熊 潇(1987-),男,硕士研究生,研究方向为分布式计算与应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130305.0817.032.html>

个决策可以抽象成任务序列和资源分配两个问题。

1.1 任务序列和资源分配

$T_i (i = 1, 2, \dots, m)$ 代表工作流中需要执行的 m 个任务, 每个任务之间可能存在约束和并发的关系。任务序列代表了工作流中所有任务组成的任务集合, 但任务前后顺序是不一定的。假设 T_j 必须在 T_i 执行完之后才能执行, 则表示 T_i 是 T_j 的先序任务, 当多个任务的先序任务都执行完毕, 且这些任务没有资源占用冲突, 则这些任务可以并发执行。需要说明的是, 任务序列的顺序并不代表任务执行顺序, 它表示的是当资源发生冲突的时候, 任务 T_i 当且仅当任务序列中 T_i 到 T_j 之间所有与任务 T_i 分配到相同资源的任务都已经执行完毕才能执行。在任务序列中, T_i 的先序任务不能出现在 T_i 之后, 否则可能造成死锁的问题, 这属于不可行任务序列, 不能保证工作流执行完毕。

$R_j (j = 1, 2, \dots, n)$ 代表提供给任务执行需要的 n 种资源。 RT_i 表示可以完成任务 T_i 的候选资源集合, RT_i 是 R 的子集。任务 T_i 运行时只需要从 RT_i 中挑选任意一种资源即可完成, 文中假设每个任务采用不同的资源完成处理的时间可以是不同的, 一种资源可以供多个任务使用, 但同一时间只能让一个任务占用。当两个任务同时竞争同一个资源时, 任务序列中靠前的任务具有占据资源的优先权。考虑到以上几点因素, 任务序列的不同和资源分配的不同会导致工作流执行时间的不同。

1.2 优化目标

文中提出的 IGA 优化的目标是, 在工作流系统资源有限, 任务可以并发运行的情况下, 计算出任务序列和资源分配的最优组合, 达到工作流完成所有任务用时最短的目的。

2 使用改进的遗传算法制定调度方案

2.1 嵌套遗传

文中所讨论的问题涉及任务序列和资源分配两组变量, 这两组变量相互影响, 如果各自独立的优化每组变量将不能够获得全局最优解, 但是采用嵌套遗传算法能够高效地搜索整个解空间, 可以克服标准遗传算法 (Canonical Genetic Algorithm, CGA) 的不足^[8], 因此文中采用了嵌套遗传算法。该算法分为两层, 第一层确定任务序列, 第二层在已知任务序列的基础上搜索最优资源分配。

为保证序列规划结果的正确性和有效性^[9,10], 引入了优先约束矩阵 (Precedence Constraint Matrix, PCM) 的概念。PCM 是一个 $M \times M$ 矩阵, 矩阵中的元素代表任务之间的关系, 文中用 C 代表优先限制矩阵, i 和 j 分别代表 PCM 矩阵的行号和列号, C_{ij} 的值代表任

务 i 和任务 j 之间的关系。

$$C_{ij} = \begin{cases} 0 & \text{任务 } i \text{ 和 } j \text{ 之间没有先后限制关系, 可以并发执行} \\ 1 & \text{任务 } j \text{ 必须在任务 } i \text{ 完成的前提下才能执行} \\ -1 & \text{任务 } i \text{ 必须在任务 } j \text{ 完成的前提下才能执行} \end{cases} \quad (1)$$

2.2 适应度函数

$$F_{\text{indi}} = M_i - T_{\text{indi}} \quad (2)$$

$$M_i = \sum_{j=1}^n \max(T_{ij}) (j = 1, 2, \dots, n) \quad (3)$$

其中 F_{indi} 代表个体的适应度, 适应度值越大代表个体越优秀。 T_{ij} 代表资源 j 执行任务 i 需要的时间, M_i 代表一个任务序列中每个任务执行所需最长时间之和, 在任务和资源已知的情况下 M_i 值是固定的, T_{indi} 代表个体执行所有任务需要的时间。

2.3 改进的遗传算法流程图

图1中变量 X 代表的是任务序列层, 变量 Y 代表资源分配层。算法主要分为两层: 任务序列层和资源分配层, 具体的改进后的遗传操作细节将在下文详细讲解。

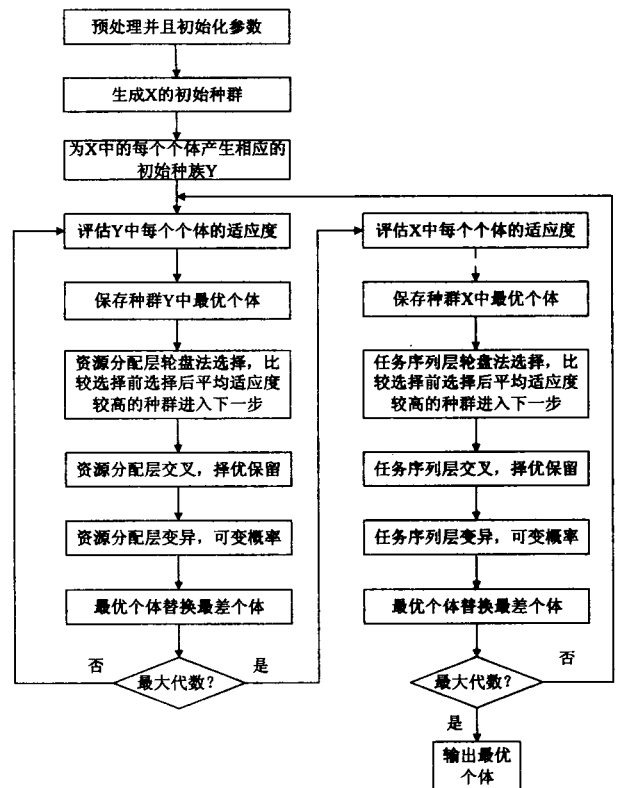


图1 改进的遗传算法流程图

2.4 任务序列层

2.4.1 染色体编码

任务序列层的染色体用可行的任务序列表示, 可行的任务序列并不要求任务是顺序执行的, 在先序任务都完成的情况下多个任务可以并发执行, 但并发执行发生资源争夺时, 将按照任务序列顺序判定资源如

何分配,但正在使用的资源不可被剥夺。

2.4.2 初始种群生成方法

随机选取可行的任务序列构成初始种群,选取可行的任务序列可以借助 PCM,每次选取的都是没有先序任务或者先序任务已经执行完毕的任务,即 C_{ij} 不等于 -1,然后把 PCM 中所有与所选取的任务相关的行和列置为 0,即选取任务 T_i 后,将 $C_{ij}(j=1,2,\dots,m)$ 和 $C_{ji}(j=1,2,\dots,m)$ 都置为 0, m 代表工作流中所有任务的数量。按此方法继续选取下一个任务,直到形成一组可行的任务序列,循环 popSize 次生成初始种群,其中 popSize 为种群的规模,随机产生的任务序列可能会存在重复。

2.4.3 精英保留操作和改进的选择操作

首先采用精英保留策略^[11]保留最优个体,然后再进行选择操作。选择操作采用轮盘赌法,并对选择前后的种群平均适应度进行比较,择优进入交叉操作。改进之后的选择操作可以防止轮盘赌法在小概率情况下选出了平均适应度还不如选择前的种群进入交叉操作。

2.4.4 改进的交叉操作

文中根据文献[12]提出适合于任务序列层的交叉操作,并能够保证交叉后能得到合法的染色体。文中提出的交叉方法在两个可行任务序列前部随机选择一个交叉点,任务序列中第一个任务到交叉点所指的任务之间的任务序列称为匹配区域。假设两个任务序列分别为父代 1,父代 2,首先将父代 1 的匹配区域加到父代 2 前面,父代 2 的匹配区域加到父代 1 前面,形成两个中间任务序列,然后移除匹配区外与匹配区相同的任务得到两个孩子序列,如图 2 所示,参考图 3 工作流程系统。

经过上述交叉过程得到的孩子个体不能保证其可行,所以需要根据 PCM 验证孩子个体的可行性。如果发现不可行,则删除任务序列中首个不可行的任务及其之后的任务,根据 PCM 生成剩余的任务,保证任务序列的可行性。为了进一步促进算法收敛,交叉完后比较两个父代和子代的适应度,适应度大的两个个体进入变异操作,适应度小的个体淘汰。

2.4.5 改进的变异操作

因为任务之间可能存在约束关系,需要保证变异后任务序列的可行性,所以变异时不能采用传统遗传算法的变异方法。文中根据文献[8]提出了适合于文中任务序列层的变异方法。首先随机选择一个染色体进行变异,随机选择任务序列中的一个任务 T_i ,删除任务序列中任务 T_i 之后的所有任务。因为使用 PCM 可以保证任务序列的可行性,所以使用 PCM 根据剩余的任务随机规划新的任务组成新的可行的任务序列,达

到变异的目的。另外,由于精英保留策略和交替方法中的择优策略会导致算法快速收敛,可能会出现算法收敛于局部最优解的情况,为了尽量使得算法能够搜索整个解空间,文中采用可变的变异概率^[13]来保持种群的多样性。

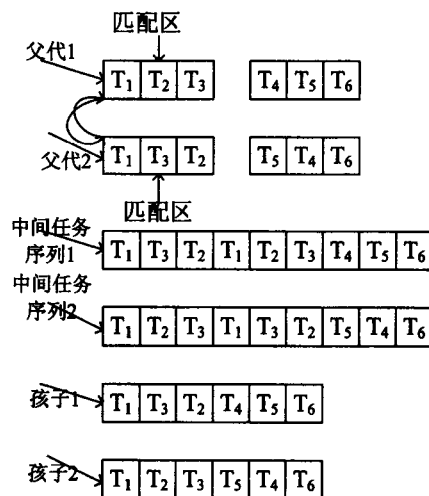


图 2 改进后的交叉方法

2.4.6 替换最差个体操作

使用精英保留下来的个体替换变异后种群中最差的个体。

2.5 资源分配层

2.5.1 染色体编码

资源分配层的染色体可以用资源序列表示,资源序列和任务序列具有对应关系,例如任务序列 T_1, T_2, T_3 , 分配给任务序列的资源序列为 R_1, R_1, R_3 , 这表示给任务 T_1 分配资源 R_1 , 给任务 T_2 分配资源 R_1 , 给任务 T_3 分配资源 R_3 。每个资源序列代表一个资源分配层个体,每个资源序列中的资源代表染色体中的基因。

2.5.2 初始种群生成操作

在任务序列已知的前提下,为每个任务随机选取一个可用资源。资源序列的长度与任务序列相同,共选取 popSize 个资源序列组成初始种群, popSize 为种群的规模。

2.5.3 精英保留操作和改进的选择操作

与任务序列层精英保留以及选择方法相同。

2.5.4 交叉操作

采用传统的单点交叉操作,随机选择交叉点进行交叉操作。

2.5.5 变异操作

随机选择变异点,在能够执行变异点所对应任务的资源集合中随机选中一种资源替换当前资源。变异概率依然如任务序列层变异操作一样变化。

2.5.6 替换最差个体操作

使用精英保留下来的最优个体替换变异后种群中最差的个体。

3 实验

3.1 实验环境和比较方法

实验硬件平台是个人笔记本电脑, CPU 为 Inter Duo 2. 10 GHz, 内存 2GB, 软件平台是 Windows XP2002, 使用 Java 开发, 开发工具为 Myeclipse7. 5。文中将比较在相同软硬件环境下, 在相同的工作流系统中, 分别采用动态方法^[4,5]、CGA 以及 IGA 方法完成所有任务需要的时间, 用时最短的方法最优秀。

3.2 实验工作流系统和实验数据

此处先用一个简单的工作流系统描述实验流程, 此处的工作流系统用 petri 网描述, 如图 3 所示。

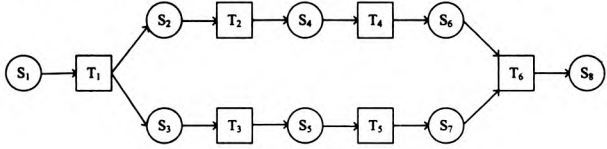


图 3 示例工作流系统

设置任务和资源之间的关系, T_i 代表任务 i , R_j 代表资源 j , T_{ij} 代表使用资源 j 完成任务 i 需要的时间, 时间用相对值表示, T_{ij} 为 0 时代表资源 j 不能用于完成任务 i , 任务和资源之间的关系如表 1 所示。

表 1 任务资源关系表

$T_i \backslash T_{ij}$	T_1	T_2	T_3	T_4	T_5	T_6
R_1	30	14	15	0	23	12
R_2	0	18	38	13	0	32
R_3	47	16	0	18	0	45
R_4	25	20	40	20	22	0

3.3 实验结果

使用动态方法^[4,5], 工作流系统分配资源 $R_4, R_1, R_2, R_3, R_4, R_1$ 去处理任务序列 $T_1, T_2, T_3, T_4, T_5, T_6$, 完成所有任务需要的时间是 97, 使用 IGA 方法工作流系统分配资源 $R_4, R_3, R_1, R_2, R_4, R_1$ 去处理任务序列 $T_1, T_2, T_3, T_4, T_5, T_6$, 完成所有任务需要的时间是 74, 使用 CGA 方法工作流系统分配资源 $R_4, R_1, R_1, R_2, R_4, R_1$ 去处理任务序列 $T_1, T_2, T_3, T_4, T_5, T_6$, 完成所有任务需要的时间是 88。从本例实验结果可以得出, 使用 IGA 方法可以使整个工作流在执行时间方面最短。随后, 用另外 9 种不同结构的工作流系统或相同结构不同的任务资源关系的工作流做了实验, 包括存在并发关系和不存在并发关系的工作流, 实验结果如图 4、图 5 所示。

实验结果表明, 在资源受限的工作流中, 当工作流中不存在并发执行的任务时, 使用 IGA 方法和动态方法执行时间相同, 且两者都优于或等于 CGA 方法, 但如果存在并发执行的任务, 使用 IGA 方法在执行时间方面总是优于或等于另外两种方法。另外, 从图 5 中

工作流实例平均执行时间可以发现, 使用 IGA 方法的工作流系统平均执行时间最短。

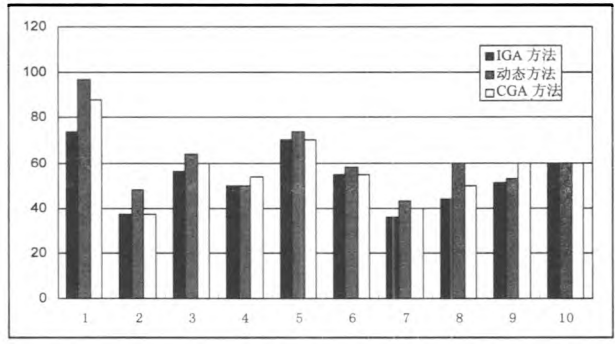


图 4 IGA 方法, 动态方法, CGA 方法时间对比

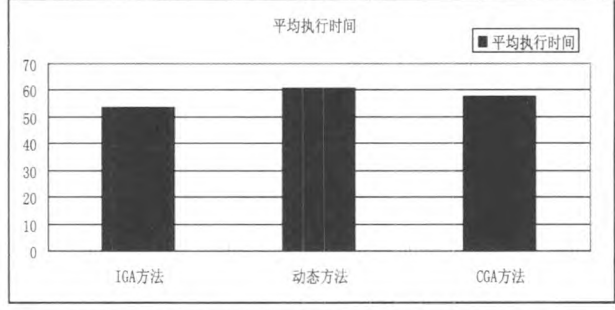


图 5 IGA 方法, 动态方法, CGA 方法平均执行时间对比

4 结束语

文中针对如何全局优化工作流系统提出了 IGA 算法, 并与基于动态调度方法和 CGA 方法的工作流系统进行了比较, 实验结果表明在资源受限的工作流系统中, 无论是否存在并发执行任务, 使用 IGA 方法会使得工作流系统在总耗时方面总是会优于或等于使用动态调度方法和 CGA 方法, 这是因为采用 IGA 算法可以对整个工作流进行全局优化, 并且在搜索范围和收敛性上做了优化。而动态调度方法只是优化了单个任务, 容易陷入局部最优, CGA 虽然也是对工作流进行全局优化, 但算法收敛性得不到保障, 得出的解可能不是最优解。

参考文献:

[1] WfMC. Terminology & Glossary[S]. [s. l.]: Workflow Management Coalition, 1999.

[2] WfMC. The Workflow Reference Model[S]. [s. l.]: Workflow Management Coalition, 1994.

[3] van der Aalst W M P, van Hee K. Workflow Management: Models, Methods, and Systems[M]. [s. l.]: MIT Press, 2002.

[4] Avanes A. Adaptive Workflow Scheduling Under Resource Allocation Constraints and Network Dynamics[J]. Proceedings of the VLDB Endowment, 2008, 1(2): 1631-1637.

[5] Lee K M. Adaptive resource scheduling for workflows considering competence and preference[C]//KES 2004. Berlin:

(下转第 115 页)

表 2 所示来揭示比较结果。突出地表明同一个规划求解问题,基本遗传算法比二进制编码的遗传算法收敛速度要慢得多。

表 2 SGA 与二进制编码遗传算法计算结果比较

问题(P)	SGA		二重结构编码遗传算法	
	最差值的 误差	最优值的 误差	最差值的 误差	最优值的 误差
P ₁	0.076	0.054	0.001	0.000
P ₂	0.128	0.112	0.020	0.000
P ₃	0.130	0.108	0.024	0.009
P ₄	0.140	0.087	0.006	0.044
P ₅	0.037	0.033	0.016	0.000
P ₆	0.145	0.082	0.035	0.000
P ₇	0.195	0.084	0.053	0.037
P ₈	0.082	0.079	0.011	0.000
P ₉	0.173	0.104	0.040	0.020
P ₁₀	0.218	0.133	0.052	0.015

4 结束语

二进制编码下的基本遗传算法在求解背包问题上展示了超强搜索能力,它具有收敛快、搜索速度快的特点,在试验中取得了比基本遗传算法、动态规划、贪心法等更好的求解效果。然而在一般情况下,旅行包问题利用基本遗传算法求解时,得到求解问题结果的近似解也不能满足逼近最优解的要求。二进制编码下的局部搜索寻优算法与基本遗传算法相结合,提高了基本遗传算法寻优方向中的导向,同时也提高了收敛速

度,又克服了传统方法容易陷入局部最优解的特点。通过计算机模拟仿真证明该算法是求解旅行包问题的有效算法,将该算法可以广泛应用于求解其它组合优化问题,比如资源分配、投资决策、科学发展观等优化问题,在并行环境中实现上述算法具有更好的近似解。

参考文献:

[1] Chatterjee S, Carrera C, Lynch A. Genetic algorithms and traveling salesman problems[J]. European Journal of Operational Research,1996,9(3):490-510.

[2] Burkowski F J. Proximity and priority applying a gene expression algorithm to the traveling salesperson problem[J]. Parallel Computing,2004,30(8):803-816.

[3] Katayama K, Sakamoto H, Narihisa H. The efficiency of hybrid mutation genetic algorithm for the traveling salesman problem [J]. Mathematical and Computer Modeling,2000,31(12):197-203.

[4] 张文修,梁怡. 遗传算法的数学基础[M]. 西安:西安交通大学出版社,2001:178-186.

[5] 曾国清. 0-1 背包问题的遗传算法求解[J]. 科技信息,2006(3):242-243.

[6] 霍红卫,许进,保铮. 基于遗传算法的 0/1 背包问题求解[J]. 西安电子科技大学学报,1999,26(4):493-497.

[7] 华福,刘晓路. 面向旅行商问题的一种改进遗传算法[J]. 计算机技术与发展,2011,21(6):51-54.

[8] 刘锐,张金波,刘蕊洁,等. 基于遗传算法求解 0-1 背包问题的算法探讨[J]. 云南民族大学学报(自然科学版),2008,17(4):377-379.

[9] 黄与林. 0-1 背包问题的贪婪算法[J]. 鄂州大学学报,2006,13(6):38-40.

[10] 余建坤,张文彬,陆玉昌. 遗传算法及其应用[J]. 云南民族学院学报(自然科学版),2002,11(4):194-197.

(上接第 111 页)

Springer-Verlag,2004:723-730.

[6] Dong Fangpeng, Akl S G. An adaptive double-layer workflow scheduling approach for grid computing[C]//Proc of the 21st International Symposium on High Performance Computing Systems and Applications. Washington DC:IEEE Computer Society,2007:156-163.

[7] Tretola G, Zimeo E. Activity pre-scheduling in grid workflows [C]//Proc of the 15th Euromicro International Conference on Parallel Distributed and Network-based Processing. Washington DC:IEEE Computer Society,2007:245-253.

[8] 朱敬华,高宏. 无线传感器网络中能源高效的任务分配算法[J]. 软件学报,2007,18(5):1198-1207.

[9] 薛俊芳,邱长华,向东. 在 Pro/E 中自动生成零件拆卸优先约束矩阵[J]. 工程图学学报,2007(3):24-29.

[10] 王歌,张林鎔,贾志新,等. 基于工程语义信息和优先约束矩阵的装配顺序规划方法[J]. 系统仿真学报,2008(19):5262-5267.

[11] de Jong K A. An Analysis of the Behavior of a Class of Genetic Adaptive Systems [D]. Michigan: University of Michigan, 1975.

[12] 梁艳春,冯大鹏,周春光. 遗传算法求解旅行商问题时的基因片段保序[J]. 系统工程理论与实践,2000(4):7-12.

[13] 丁华福,刘晓路,唐远新,等. 面向旅行商问题的一种改进遗传算法[J]. 计算机技术与发展,2011,21(6):51-54.

基于改进遗传算法的工作流调度研究


作者:

王晓军, 熊潇, WANG Xiao-jun, XIONG Xiao

作者单位:

南京邮电大学信息网络技术研究所, 江苏南京, 210003

刊名:

计算机技术与发展 

英文刊名:

Computer Technology and Development

年, 卷(期):

2013, 23(7)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201307027.aspx