

项目优化调度的病毒协同进化遗传算法*

胡仕成¹⁺, 徐晓飞¹, 李向阳²

¹(哈尔滨工业大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

²(哈尔滨工业大学 管理学院, 黑龙江 哈尔滨 150001)

A Virus Coevolution Genetic Algorithm for Project Optimization Scheduling

HU Shi-Cheng¹⁺, XU Xiao-Fei¹, LI Xiang-Yang²

¹(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

²(School of Management, Harbin Institute of Technology, Harbin 150001, China)

+Corresponding author: Phn: +86-451-6419787, E-mail: hu_shicheng@yahoo.com.cn, <http://www.cs.hit.edu.cn>

Received 2002-11-12; Accepted 2003-03-24

Hu SC, Xu XF, Li XY. A virus coevolution genetic algorithm for project optimization scheduling. *Journal of Software*, 2004,15(1):49~57.

<http://www.jos.org.cn/1000-9825/15/49.htm>

Abstract: In this paper, a virus coevolution genetic algorithm (multi-mode project scheduling-virus co-evolution genetic algorithm, MPS-VEGA) for the precedence and resource constrained multi-mode project scheduling problem is presented, and the encoding of the solution and the operators such as selection, crossover, mutation and virus_infection are given. MPS-VEGA is used to obtain the optimal scheduling sequences and resource modes for the activities of the project so that the project cost is minimized, which can transmit evolutionary genes not only between parent and child generations vertically by the genetic operators but also in the same generation horizontally by the virus_infection operator so as to perform a global search and a local search, respectively. The schema theorem is adopted to analyze the performance of MPS-VEGA. The theoretical analysis and experimental results show that the MPS-VEGA outperforms the GA. For the multi-mode project scheduling problem with different optimization objectives, MPS-VEGA can simultaneously give standard the optimal scheduling sequences subject to the precedence constraints and the optimal resource modes for the activities of the project.

Key words: resource-constrained project scheduling; multi-mode; cost optimization; virus evolution; genetic algorithm

摘要: 针对次序约束和资源约束的多模式项目调度问题提出了一种病毒协同进化遗传算法,并提出了了解的

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.863-511-944-001, 2001AA414 010 (国家高技术研究发展计划(863)); the Key Science-Technology Project of the National 'Tenth Five-Year-Plan' of China under Grant No.2001BA201A03 (国家“十五”重点科技攻关项目)

作者简介: 胡仕成(1970—),男,湖北浠水人,博士生,主要研究领域为 CIMS,管理与决策信息系统;徐晓飞(1962—),男,教授,博士生导师,主要研究领域为 CIMS,数据库,管理与决策信息系统;李向阳(1950—),男,教授,博士生导师,主要研究领域为 CIMS,技术经济,管理与决策信息系统。

编码、选择、交叉、变异和病毒感染操作等.算法用于求解项目活动的一个最优调度顺序和资源模式以使项目的成本最低,其操作特点是既可以通过遗传操作在父子代群体之间纵向传播进化基因进行全局搜索,又可以通过病毒感染操作在同一代群体内横向传播进化基因进行局部搜索.利用模板理论对算法的性能进行了分析.理论分析和实验结果表明,算法的搜索性能优于一般的遗传算法.算法对于不同优化目标的多模式项目调度问题可以同时求得一个满足次序约束的项目活动的最优调度顺序和满足资源约束的最优资源模式.

关键词: 资源约束项目调度;多模式;成本优化;病毒进化;遗传算法

中图法分类号: TP18 文献标识码: A

在制造企业的生产计划、经营计划和项目管理中,项目调度(project scheduling,简称 PS)是一个重要的优化问题,PS 需要解决在满足一定的约束条件下达到某种最优的目标,如工期最短、成本最小等,从而满足企业的某种经营目的,同时,很多组合优化问题都可以归结为 PS 的特例,如车间作业调度^[1]等,因此研究 PS 具有广泛的实用价值和理论意义.PS 的约束包括活动之间的先后次序约束和资源限量约束,资源包括可重复使用的资源和不可重复使用的资源^[2],每个活动的运作模式也可以有多种(multi-mode)^[3],这样的 PS 称为 MPS(multi-mode project scheduling),PS 是一个 NP-hard 问题,有两种以上的不可重复使用的资源约束的 MPS 是一个 NP-complete^[4]问题.Kolisch^[5]提出了比较完整的求解 PS 的串行和并行的理论和方法,Kolisch^[6]等人给出了求解 PS 的局部搜索算法,Reych^[7]采用了分支定界方法对 PS 进行求解,Fayez^[8]和 Taeho^[9]等人对 MPS 进行了研究,并给出了启发式求解方法,Joanna^[10]采用了模拟退火算法对 MPS 进行求解.以上算法的研究仅限于以时间为目标的 PS,对企业来说,成本也是一个重要的经营目标,因此对以成本为目标的 PS 的研究同样具有重要意义.

根据病毒进化理论^[11],病毒是一种特有的生物,具有一种特有的感染功能,它能获得一个个体的染色体基因,并且感染给另一个个体,使得该个体的部分染色体基因发生相应的变化,从而改变该个体的遗传信息,这种遗传信息又通过遗传传递给下一代,从而大大加速了生物的进化换代.1996年,Kubota 提出了基于病毒进化理论的遗传算法 VEGA(virus co-evolution genetic algorithm),并成功地用于求解旅行商问题^[11]、自组织制造系统的调度问题^[12]、弹道规划问题^[13]、背包问题^[14]等,显示了求解 NP-hard 优化问题的优越性.VEGA 在进化计算过程中产生两种群体:主群体和病毒群体.主群体对应问题的解空间,进行 GA 的遗传操作,在上下代群体之间纵向传递进化基因,实施解空间的全局搜索.病毒群体进行病毒感染操作,在同代个体之间横向传递进化基因,实施解空间的局部搜索,VEGA 将主群体的全局进化和病毒群体的局部进化进行动态结合,从而快速得到问题的全局近似最优解.本文将针对以成本为目标的 MPS 提出一种病毒协同进化遗传算法 MPS-VEGA(multi-mode project scheduling-virus co-evolution genetic algorithm).

1 问题描述

一个 MPS 可以描述为:活动集 $V = \{0, 1, \dots, n+1\}$, 0 和 $n+1$ 分别表示虚拟的起始活动和结束活动;活动 $j(j \in V)$ 有多个模式 $M_j = \{1, \dots, |M_j|\}$, $M_0 = M_{n+1} = \{1\}$;活动 j 在模式 m_j 下的提前期 $p_{j,m_j} \in Z^+(m_j \in M_j)$, $p_{0,1} = p_{n+1,1} = 0$;第 $k(k \in R^\rho)$ 种可重复使用的资源(称为资源 ρ)单位时间限量为 R_k^ρ ,单位成本为 c_k^ρ ,活动 j 在模式 m_j 下所消耗的资源量为 $r_{m_j,k}^\rho$, $r_{0,k}^\rho = r_{n+1,k}^\rho = 0$;第 $k(k \in R^v)$ 种不可重复使用的资源(称为资源 v)限量为 R_k^v ,单位成本 c_k^v ,活动 j 在模式 m_j 下所消耗的资源量为 $r_{m_j,k}^v$, $r_{0,k}^v = r_{n+1,k}^v = 0$.如何确定活动的调度顺序 $J = (j_0, \dots, j_{n+1})(j_0 = 0, j_{n+1} = n+1)((j_0, \dots, j_{n+1})$ 为 $(0, 1, \dots, n+1)$ 的一个排列),模式 $M = (m_{j_0}, \dots, m_{j_{n+1}})$ 和开始时间 $S = (S_{j_0}, \dots, S_{j_{n+1}})(S_0 = 0, S_{n+1} = \text{项目工期})$,使得项目活动满足次序约束关系,单位时间内资源 ρ 限量约束和项目工期内资源 v 限量约束的条件下项目的成本最优.按照 Brucker^[15]的分类表示方法问题可表示为 $MPS|prec|\sum C_j$, $MPS|prec|\sum C_j$ 在于求得活动的最优调度顺序和资源模式,这是一个 NP-hard 问题,当 $|R^v| \geq 2$ 时是一个 NP-complete^[4]问题,当活动的模式向量 M 一旦确定,活动 j 的模式 μ 便确定了活动的提前期、资源 ρ 和资源 v 的消耗量,单位资源的成本是一定的,活动 j 的成本也便确定:

$$C_j = C_{j\mu} = [C_{j\mu}^\rho] + (C_{j\mu}^v) = \left[\sum_{t=S_j}^{S_j+p_{j\mu}} \left(\sum_{k \in R^\rho} c_k^\rho r_{j\mu k}^\rho \right) \right] + \left(\sum_{k \in R^v} c_k^v r_{j\mu k}^v \right),$$

则项目的成本 $\sum C_j$ 也随之确定,同时 $MPS|prec|\sum C_j$ 退变为经典的 $PS|prec|C_{\max}$ (工期最小的单模式项目调度问题).

定义 1. 如果 $J(M$ 或 $S)$ 满足次序约束、单位时间内资源 ρ 限量约束或项目工期内资源 v 限量约束,则称 $J(M$ 或 $S)$ 次序可行,资源 ρ 可行或资源 v 可行,记为 $J(M$ 或 $S) \in \Psi_T, J(M$ 或 $S) \in \Psi_\rho$ 或 $J(M$ 或 $S) \in \Psi_v$. 如果 J, M 和 $S \in \Psi_T \cap \Psi_\rho \cap \Psi_v$, 并且 M 和 S 对应于同一个 J , 则称 (J, M, S) 为 $MPS|prec|\sum C_j$ 的一个调度, 记为 $(J, M, S) \in \Psi_T \cap \Psi_\rho \cap \Psi_v$.

定义 2. 向量的两种运算,连接 \cup 和裁剪 \setminus .

设有向量 $w = (w_1, w_2, w_3, w_4, w_5)$ 和 $v = (v_1, v_2, v_3, v_4, v_5)$, 则:

- (1) $w \cup v = (\overbrace{w_1, w_2, w_3, w_4, w_5}^w, \overbrace{v_1, v_3, v_5}^{v-w \cap v})$, 称为 v 连接到 w ;
- (2) $v \cup w = (\overbrace{v_1, v_2, v_3, v_4, v_5}^v, \overbrace{w_1, w_3, w_5}^{w-v \cap w})$, 称为 w 连接到 v ;
- (3) $w \setminus v = (\overbrace{w_1, w_3, w_5}^{w-w \cap v})$, 称为 w 按 v 裁剪;
- (4) $v \setminus w = (\overbrace{v_1, v_3, v_5}^{v-v \cap w})$, 称为 v 按 w 裁剪.

显然,向量的这两种运算和集合对应的运算不同在于:向量运算后仍然是一个向量,且各分量仍然保持在运算前原向量中的偏序关系,而集合运算后是一个集合.一个向量可以用来表示 MPS 活动的一个调度顺序,因此向量的这两种运算的意义是在交叉操作过程中保持活动之间的次序约束关系,通过交叉操作产生 MPS 活动的一个最优的调度顺序.

2 病毒协同进化遗传算法 MPS-VEGA

2.1 染色体编码方案

2.1.1 主个体染色体编码

主个体(host)染色体编码由两部分组成:

$$I = \begin{pmatrix} J \\ M \end{pmatrix} = \begin{pmatrix} j_0 & \dots & j_{n+1} \\ m_{j_0} & \dots & m_{j_{n+1}} \end{pmatrix}.$$

J 表示 MPS 活动的一个调度顺序, M 表示 MPS 活动对应的资源模式向量, J 和 M 中的基因分别称为活动基因和模式基因. 当确定了活动的调度顺序 J 和对应的模式向量 M 后, 采用串行调度算法^[5], 活动的开始时间向量 S 不难惟一确定, 因此每个染色体惟一地对应了一个调度 (J, M, S) .

2.1.2 病毒个体染色体编码

由于模式向量 M 确定了项目的成本, 因此只对模式向量 M 产生病毒个体并进行病毒进化操作. 病毒个体(virus)编码为

$$H = (m_0, m_1, \dots, m_{n+1}).$$

H 是对应于调度顺序 $(0, 1, \dots, n+1)$ 的模式向量. 病毒个体产生于主个体的模式向量 M , 是模式向量 M 的字符串, 串中包含通配符*, 通配符*不表示任何模式. 串中除通配符以外的字符称为有效字符, 有效字符表示一个活动的具体资源模式, 因此病毒个体表示的是 MPS 部分活动的模式. 如 $H_1 = (*, 3, *, 2, *, *)$ 表示 $m_1 = 3, m_3 = 2$ 是产生于下面主个体的病毒个体

$$I = \begin{pmatrix} 0 & 3 & 1 & 4 & 2 & 5 \\ 1 & 2 & 3 & 7 & 9 & 1 \end{pmatrix}.$$

2.2 适应度

2.2.1 主个体的适应度函数

主个体 I 的适应度函数为

$$fithost = \begin{cases} (\sum C_j)(I) + C_{\max}(I) & I \in \Psi_v \\ B + T + \pi R^v(M) & \text{else} \end{cases},$$

其中, $(\sum C_j)(I)$ 和 $C_{\max}(I)$ 分别是染色体 I 对应的项目成本和项目工期, B 和 T 分别是项目的最大成本和最长工期, $\pi R^v(M)$ 为模式向量 M 对资源 v 的超出额度:

$$\pi R^v(M) = \sum_{\substack{k \in R^v \\ \pi R_k^v(M) < 0}} |\pi R_k^v(M)|,$$

其中 $\pi R_k^v(M) = R_k^v - \sum_{j \in V} r_{j_m, k}^v$.

由于 $(\sum C_j)(I) \gg C_{\max}(I)$, 因此适应度可以保证在目标成本大小相差不大的情况下选择时间较优的调度, 并且资源 v 不可行的染色体的适应度总是大于资源 v 可行的染色体的适应度. 在资源 v 不可行的染色体中, 超出额度较大者的适应度较大. 适应度体现了 MPS 的一个解(包括活动的调度顺序 J 和资源模式向量 M) 的优劣程度, 其中项目工期 $C_{\max}(I)$ 由活动的调度顺序 J 唯一确定(由串行调度算法^[5]确定各个活动的开始时间(S_{n+1} = 项目工期)), 因此它体现了调度顺序 J 的优劣程度, 而 $(\sum C_j)(I)$ 直接体现了资源模式向量 M 的优劣程度.

2.2.2 病毒个体的适应度函数

一个病毒个体可以感染若干个主个体, 病毒个体的适应度用它所感染的主个体感染前后的适应度的变化来表示. 假设病毒 i 所感染的主个体的集合为 U , U 中主个体 l 被感染前后的适应度函数分别表示为 $fithost_l$ 和 $fithost'_l$, 则病毒 i 的适应度函数为

$$fitvirus_i = \sum_{l \in U} (fithost'_l - fithost_l).$$

U 中的一个主个体 l 表示的是 MPS 解空间中的一个解, 其适应度体现了解的优劣程度, 因此, 病毒个体的适应度体现了部分活动的模式对 MPS 解空间中的多个解的进化计算效果.

病毒的生存时间用生命力(life)来表现, 第 $t+1$ 代的病毒个体 i 的生命力表示如下:

$$life_{i,t+1} = \gamma \times life_{i,t} - fitvirus_i,$$

其中 γ 为生命力递减率, 如果 $life_{i,t} < 0$, 说明该部分活动的模式已不能对 MPS 的解产生进化效果, 则需要产生 MPS 新的部分活动的模式.

2.3 进化操作

2.3.1 主个体的遗传操作 GA

初始群体的产生(initialization): 通过以下步骤产生规模为偶数 POP_{host} 的初始群体 $hostpop(t)$.

首先, 为每个活动 j_i 选择模式 $m_{j_i}^*$, $m_{j_i}^*$ 为 j_i 所有模式中消耗资源 v 数量最少的一种, 从而生成一个资源 v 可行的第 1 个模式向量; 接着反复执行以下步骤直到生成 $pop_{\text{host}}-1$ 个资源 v 可行的模式向量: 随机选择一个活动 j_i ; 接着, 反复从其模式集 M_{j_i} 中随机选择一个模式 m'_{j_i} ($m'_{j_i} \neq m_{j_i}^*$) 替换 $m_{j_i}^*$, 直到生成的新的模式向量满足资源 v 约束.

交叉(crossover): 每一对父代个体通过交叉操作产生一对子代个体.

设父代中被选择用于交叉操作的两个个体分别是

$$I^M = \begin{pmatrix} J^M \\ M^M \end{pmatrix} = \begin{pmatrix} j_0^M & \dots & j_{n+1}^M \\ m_{j_0^M}^M & \dots & m_{j_{n+1}^M}^M \end{pmatrix}, I^F = \begin{pmatrix} J^F \\ M^F \end{pmatrix} = \begin{pmatrix} j_0^F & \dots & j_{n+1}^F \\ m_{j_0^F}^F & \dots & m_{j_{n+1}^F}^F \end{pmatrix}.$$

首先从 $\{1, \dots, n\}$ 中产生两个随机整数 p 和 q , 以下运算 \ 和 \cup 满足定义 2, 令 $J_1^M = (j_0^M, \dots, j_p^M)$, $J_1^F = (j_0^F, \dots, j_p^F)$, 则

$$J_1^D = J_1^M, J_2^D = J^F \setminus J_1^M, J^D = J_1^D \cup J_2^D,$$

$$J_1^S = J_1^F, J_2^S = J^M \setminus J_1^F, J^S = J_1^S \cup J_2^S;$$

$$M_1^D = (m_{j_0^D}^M, \dots, m_{j_q^D}^M), M_2^D = (m_{j_{q+1}^D}^F, \dots, m_{j_{n+1}^D}^F), M^D = M_1^D \cup M_2^D,$$

$$M_1^S = (m_{j_0^S}^F, \dots, m_{j_q^S}^F), M_2^S = (m_{j_{q+1}^S}^M, \dots, m_{j_{n+1}^S}^M), M^S = M_1^S \cup M_2^S.$$

其中, $m_{j_i^D}^M (m_{j_i^D}^F)$ 表示 $I^M (I^F)$ 中活动 j_i^D 对应的模式, $m_{j_i^S}^M (m_{j_i^S}^F)$ 表示 $I^M (I^F)$ 中活动 j_i^S 对应的模式, 则 I^M 和 I^F 产生的两个子代个体分别是

$$I^D = \begin{pmatrix} J^D \\ M^D \end{pmatrix}, I^S = \begin{pmatrix} J^S \\ M^S \end{pmatrix}.$$

交叉操作的作用是,在 MPS 的解空间中随机产生活动的新的调度顺序和资源模式。

变异(mutation):设 P_{j_i} 表示活动 j_i 的所有前行活动的集合,对于每个活动 $j_i (0 < i < n)$,如果 $j_i \notin P_{j_{i+1}}$,则以概率 p_{mut} 发生变异: j_i 和 j_{i+1} 换位;每个模式 $m_{j_i} (0 < i < n+1)$ 以概率 p_{mut} 发生变异:从 M_{j_i} 中随机选择一个模式 $m'_{j_i} (m'_{j_i} \neq m_{j_i})$ 替换 m_{j_i} 。

变异操作使得 MPS 活动的部分调度顺序和资源模式发生变化从而产生活动的新的调度顺序和资源模式。

选择(selection):为了使当前最优解保持下去,采用排序法(ranking)^[16],即对遗传操作前后的所有个体按适应度值进行排序,使最优的个体保持下去。

选择的作用是,从 MPS 已有的解集中选择出较优的调度顺序和资源模式。

2.3.2 病毒个体的进化操作 VE

病毒感染(virus_infection):利用病毒染色体基因替换主染色体中相应的模式基因,从而产生新的主个体,如图 1 所示.每个病毒个体以概率 p_{infect} 感染每个主个体,如果某主个体感染后的适应度减小了,则用感染后的主个体替换感染前的主个体.因此,病毒感染操作的作用在于,用部分活动的资源模式替换 MPS 的多个解的活动的资源模式,如果替换后的解优于替换前的解,则用替换后的解代替替换前的解。

复制(copy):随机选择一个主个体,主染色体每个模式基因以概率 p_{copy} 代换病毒染色体中相应基因,从而产生新的有效字符更多的病毒个体,如图 2 所示.初始病毒个体也是通过复制产生的,即通过一个主染色体以概率 p_{copy} 复制到一个“无感染能力(infectless)”的病毒个体,如图 3 所示.复制操作的作用在于,用 MPS 一个解的部分活动的资源模式替换已有的一个部分活动的资源模式,从而增加其进化计算能力,或者产生一个新的部分活动的资源模式。

删减(cut):病毒染色体每个基因以概率 p_{cut} 用通配符代换,如图 4 所示.删减操作的作用在于,减少一个已有的一个部分活动的资源模式的数量,从而降低其进化计算能力。

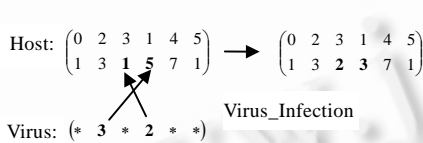


Fig.1 Virus_Infection operator
图 1 病毒感染操作

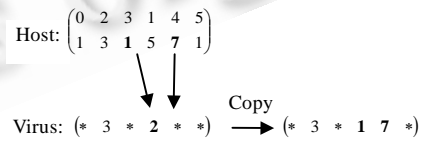


Fig.2 Copy operator for virus
图 2 病毒复制操作

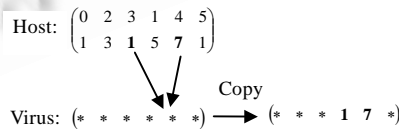


Fig.3 Copy operator for initial virus
图 3 初始病毒复制操作

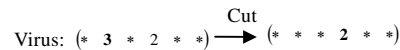


Fig.4 Cut operator for virus
图 4 病毒删减操作

2.4 病毒进化遗传算法MPS-VEGA

MPS-VEGA 可以描述如下:

参数初始化: $t := 0, POP_{\text{host}}, POP_{\text{virus}}, p_{\text{copy}}, p_{\text{cut}}, p_{\text{cross}}, p_{\text{mut}}, p_{\text{infect}}, p_{\text{maxinfect}}, \mathcal{V}$;

群体初始化: 执行 Initialization 操作产生 POP_{host} 个个体 j 的初始主群体 $\text{hostpop}(t)$, 计算 $\text{fit}_{\text{host}j}$; 执行 Copy 操作产生 POP_{virus} 个个体 i 的初始病毒群体 $\text{viruspop}(t)$, 初始化相关参数, $\alpha_{i,t} := 0, \text{life}_{i,t} := 0$;

//以下步骤执行 GA 操作.

Crossover: 由 $\text{hostpop}(t)$ 产生 $\text{crosshostpop}(t+1)$;

Mutation: 由 $\text{crosshostpop}(t+1)$ 产生 $\text{muthostpop}(t+1)$, 计算每个主个体 j 的适应度 $\text{fit}_{\text{host}j}$;

Selection: 由 $\text{hostpop}(t) \cup \text{muthostpop}(t+1)$ 产生 POP_{host} 个个体的 $\text{hostpop}(t+1)$;

//以下步骤执行病毒的进化操作 VE, 对 $\text{viruspop}(t)$ 中每个病毒个体 i 执行以下操作.

Virus_Infection: 病毒个体 i 以概率 $p_{\text{infect}} = \min\{(1 + \alpha_{i,t}) p_{\text{infect}}, p_{\text{maxinfect}}\}$ 感染 $\text{hostpop}(t+1)$ 中每个主个体, 记 $\text{hostpop}(t+1)$ 中被病毒 i 感染后的子群体为 U ; 计算 U 中每个个体 l 的适应度 $\text{fit}_{\text{host}l}'$, 如果 $\Delta \text{fit}_{\text{host}l} = \text{fit}_{\text{host}l}' - \text{fit}_{\text{host}l} < 0$, 则用个体 l 替换 $\text{hostpop}(t+1)$ 中相应的个体; 计算 $\text{fit}_{\text{virus}i}$ 和 $\alpha_{i,t+1} := \alpha_{i,t} - \text{fit}_{\text{virus}i} / \sum_{l \in U} \text{fit}_{\text{host}l}$;

Copy 或 Cut: 如果 $\text{fit}_{\text{virus}i} < 0$, 则从 U 中随机选择一个主个体对病毒个体 i 执行 Copy 操作; 否则对病毒个体 i 执行 Cut 操作, 将病毒个体 i 记入 $\text{viruspop}(t+1)$; 计算 $\text{life}_{i,t+1}$;

产生新病毒: 如果 $\text{life}_{i,t+1} < 0$, 则从 $\text{hostpop}(t+1)$ 中随机选择一个主个体执行 Copy 操作, 产生新的病毒个体, 替换 $\text{viruspop}(t+1)$ 中的病毒个体 i , 并初始化相关参数, $\alpha_{i,t+1} := 0, \text{life}_{i,t+1} := 0$;

结束条件: $t := t + 1$, 若满足停止条件, 则从 $\text{hostpop}(t)$ 中选择适应度最小的个体进行解码得到最优解.

MPS-VEGA 由 GA 和 VE 两部分组成, GA 进行 MPS 最优解(包括活动的调度顺序和资源模式)的全局搜索, 通过 Crossover 和 Mutation 在 MPS 的全局解空间中随机搜索活动的调度顺序和资源模式, 通过 Selection 产生 MPS 解空间中较优的调度顺序和资源模式; VE 进行 MPS 最优解的局部搜索, 通过 Virus_Infection 在 MPS 的解空间中产生多个部分活动具有相同的较优资源模式的解(包括活动的调度顺序和资源模式), 通过 Copy 可以增加这些解中具有相同较优资源模式的活动的数目, 通过 Cut 可减少具有相同资源模式的活动的数目. 整个 MPS-VEGA 是一个动态搜索过程, 如果调度顺序和资源模式较优, 则通过 Copy 逐渐加速它的产生过程; 如果非优, 则通过 Cut 逐渐减缓它的产生过程, 因此相对于 GA, VEGA 可以极大地加快 MPS 的最优调度顺序和资源模式的产生过程.

2.5 病毒进化遗传算法的性能分析

模板理论是分析进化算法性能的一种重要方法, 病毒进化遗传算法在本质上也是一种进化算法, 因此其进化计算的速度在于以较快的速度产生较多的模板的期望个数.

首先定义主个体 I 的另一种适应度函数如下:

$$\text{fitness}(I) = B + T + \sum_{k \in R^0} R_k^0 - \text{fit}_{\text{host}}(I).$$

这样每个主个体就可以采用轮盘赌方式进行选择, 由病毒个体的编码方法可知, 一个病毒个体实质上表示的是一个模板(Schema) H^* . 设 $T(H^*, \text{hostpop}(t))$ 表示在群体 $\text{hostpop}(t)$ 中具有模板 H^* 的主个体的集合, 经过 GA 的进化计算后, 在 $\text{hostpop}(t+1)$ 中具有模板 H^* 的主个体的期望数^[16]为

$$E(H^*, t+1) = f(H^*, t)N(H^*, t),$$

其中, $N(H^*, t)$ 表示 $T(H^*, \text{hostpop}(t))$ 所包含的主个体数, 即 $N(H^*, t) = |T(H^*, \text{hostpop}(t))|$,

$$f(H^*, t) = \frac{\sum_{I: I \in T(H^*, \text{hostpop}(t))} \text{fitness}(I)}{\sum_{I: I \in \text{hostpop}(t)} \text{fitness}(I)} = \frac{|T(H^*, \text{hostpop}(t))|}{|T(H^*, \text{hostpop}(t))|} \times \frac{\sum_{I: I \in T(H^*, \text{hostpop}(t)) \cup U} \text{fitness}(I) + \sum_{I: I \in U} \text{fitness}(I)}{\sum_{I: I \in \text{hostpop}(t) \cup U} \text{fitness}(I) + \sum_{I: I \in U} \text{fitness}(I)},$$

其中 U 为被病毒 H^* 感染的主个体的集合. 所以,

$$\begin{aligned}
 E(H^*, t+1) &= |\text{hostpop}(t)| \times \frac{\sum_{I:I \in T(H^*, \text{hostpop}(t)) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}(I)}{\sum_{I:I \in \text{hostpop}(t) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}(I)} \\
 &= POP_{\text{host}} \times \frac{\sum_{I:I \in T(H^*, \text{hostpop}(t)) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}(I)}{\sum_{I:I \in \text{hostpop}(t) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}(I)}.
 \end{aligned}$$

如果再通过病毒 H^* 进行进化计算后,则在 $\text{hostpop}(t+1)$ 中具有模板 H^* 的主个体的期望数为

$$E'(H^*, t+1) = POP_{\text{host}} \times \frac{\sum_{I:I \in T(H^*, \text{hostpop}(t)) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}'(I)}{\sum_{I:I \in \text{hostpop}(t) \cup U} \text{fitness}(I) + \sum_{I:I \in U} \text{fitness}'(I)}.$$

由于 $\forall I \in U, \text{fitness}'(I) \geq \text{fitness}(I) > 0$, 所以,

$$E'(H^*, t+1) \geq E(H^*, t+1).$$

这就是说,VEGA 能够比 GA 以更快的速度产生较多的模板的期望个数.

3 实验结果

随机产生活动个数依次为 $N=10,20,30,40,50,60,70,80,90,100$ 的 MPS 问题 10 个,对每个问题又随机生成 10 个实例,对这 100 个实例分别进行 GA,VE 和 VEGA 的计算,采用 3 种指标评价它们的算法性能:每个问题的 10 个实例达到最优值的平均时间(avg sec)、平均计算代数(avg gen)以及实例的最优值占最大值(B)的百分比的平均值(avg. opt).算法的参数值初始化为 $POP_{\text{host}}=100, p_{\text{mut}}=0.05, POP_{\text{virus}}=10, p_{\text{copy}}=0.2, p_{\text{cut}}=0.15, p_{\text{infect}}=0.02, p_{\text{maxinfect}}=0.1, \gamma=0.9$.采用 MATLAB6.0 在 1GhzCPU 单机上计算,结果如图 5~图 7 所示.从图 5、图 6 可以看出:相对于 GA,VEGA 能够以较短的计算时间和较少的计算代数产生 MPS 的最优解(包括活动的调度顺序和资源模式),说明利用部分活动的资源模式(即病毒个体)对 MPS 的可行解(即主个体)进行 VE 进化计算能够加速最优解的产生过程;从图 7 可以看出:仅利用部分活动的资源模式(即病毒个体)对 MPS 的可行解(即主个体)进行 VE 进化计算不能产生最优解,只能产生次优解.由此可知:VE 有很强的局部搜索性能,能够加快 MPS 的求解收敛速度,但不能产生 MPS 的最优解;GA 有很强的全局搜索能力,能产生 MPS 的最优解;VEGA 有很强的全局搜索和局部搜索能力,能够以较快的速度产生 MPS 的最优解.

在求解 MPS 问题时,参数值对 VEGA 的求解效果有很大影响,如病毒群体规模(POP_{virus})和病毒的最大感染率($p_{\text{maxinfect}}$)等.保持主群体规模不变($POP_{\text{host}}=100$),取 $POP_{\text{virus}}=6,8,10,12,14,16, p_{\text{maxinfect}}=0.2,0.13,0.1,0.07,0.03$,以 100 个实例的平均最优值(avg opt)为评价指标,计算结果如图 8 所示.结果显示,当病毒规模较小时,应适当增加 $p_{\text{maxinfect}}$ 的值才能求得 MPS 的最优值,但 $p_{\text{maxinfect}}$ 太大又容易受限于局部最优;增加病毒群体的规模有助于搜索到 MPS 的最优解,但计算代价太大.结果表明,对于 MPS,当参数取值为 $POP_{\text{virus}}=POP_{\text{host}}/10, p_{\text{maxinfect}}=0.1$ 时,VEGA 能以较快的速度产生最优的调度顺序和资源模式.

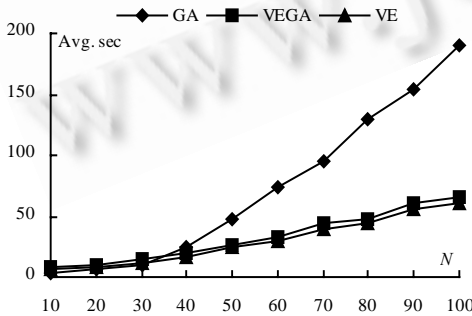


Fig.5 Curves of avg. sec changing with N
图 5 平均计算时间随 N 变化的曲线

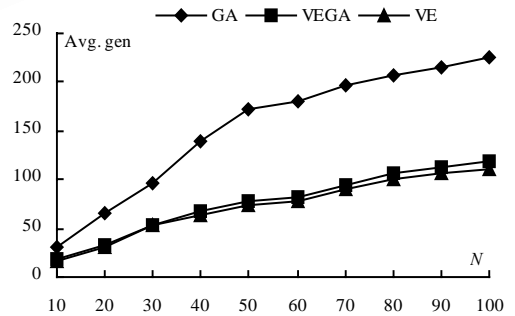


Fig.6 Curves of avg. gen changing with N
图 6 平均计算代数随 N 变化的曲线

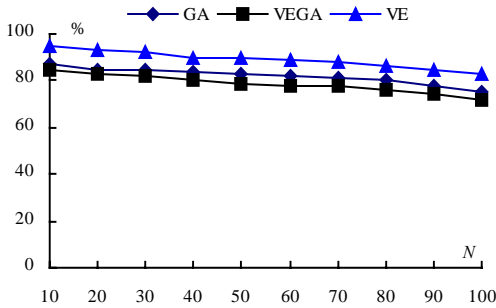


Fig.7 Curves of avg. opt changing with N
图7 平均最优值随 N 变化的曲线

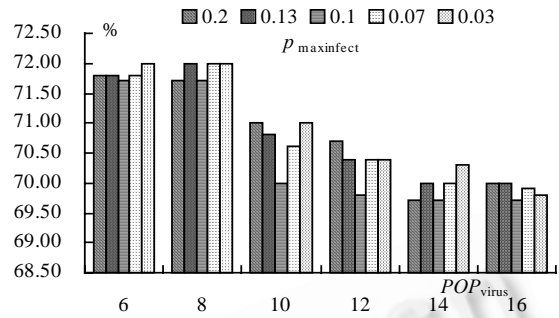


Fig.8 Effects of virus population size and max infection rate
图8 病毒群体大小和最大感染概率的影响

4 结论

本文对于一种以成本为优化目标的多模式项目调度问题 MPS 进行了研究,提出了这一问题的优化模型,并针对这一优化求解问题提出了一种病毒进化遗传算法 MPS-VEGA. MPS-VEGA 利用了 GA 的全局搜索和 VE 的局部搜索动态结合的搜索性能,从而能以很好的收敛速度求得 MPS 的最优解,即活动的最优调度顺序和资源模式的最佳方案.但是 VE 只是一种局部搜索,因此 VE 单独不能得到 MPS 的最优解.另外, MPS-VEGA 的收敛速度、收敛效果与病毒群体规模及最大感染率等参数值也密切相关.本文所提出的 MPS-VEGA 为 MPS 这一调度难题求解最优调度顺序和资源模式提供了一种新的有效方法,同时对于研究不同优化目标的 MPS 具有一定的理论贡献和实际应用价值.

References:

- [1] Sprecher A, Kolisch R, Drexel A. Semi-Active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 1995,80(1):94~102.
- [2] Nudtasomboon N, Randhawa SU. Resource-Constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs. *Computers & Industrial Engineering*, 1997,32(1):227~242.
- [3] Sprecher A, Drexel A. Multi-Mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 1998,107(2):431~450.
- [4] Roland H. Resource-Constrained project scheduling: A heuristic for the multi-mode case. *OR-Spektrum*, 2001,23(3):335~357.
- [5] Kolisch R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 1996,90(2):320~333.
- [6] Kolisch R, Drexel A. Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 1997,29(11):987~999.
- [7] De Reyck B, Herroelen W. A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 1998,111(1):152~174.
- [8] Boctor FF. A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. *European Journal of Operational Research*, 1996,90(2):349~361.
- [9] Taeho A, Selcuk ES. The resource constrained project scheduling problem with multiple crashable modes: A heuristic procedure. *European Journal of Operational Research*, 1998,107(2):250~259.
- [10] Józefowska J, Mika M, Róć-Bzycki R, Waligóra G, Wglarz J. Simulated annealing for multi-mode resource-constrained project scheduling. *Annals of Operations Research*, 2001,102(1/4):137~155.
- [11] Naoyuki K, Koji S, Fukuda T. The role of virus infection in virus-evolutionary genetic algorithm. In: Proc. of the IEEE Int'l. Conf. on Evolutionary Computation. Nagoya: IEEE, 1996. 182~187.
- [12] Kubota N, Arakawa T, Fukuda T, Shimojima K. Fuzzy manufacturing scheduling by virus-evolutionary genetic algorithm in self-organizing manufacturing system. In: Proc. of the 6th IEEE Int'l. Conf. on Fuzzy Systems. Barcelona: IEEE, 1997. 1283~1288.

- [13] Kubota N, Arakawa T, Fukuda T, Shimojima K. Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm. In: Proc. of the 1997 IEEE Int'l. Conf. on Robotics and Automation. Albuquerque: IEEE, 1997. 205~210.
- [14] Kubota N, Fukuda T. Schema representation in virus-evolutionary genetic algorithm for knapsack problem. In: Proc. of the 1998 IEEE World Congress on Computational Intelligence, The 1998 IEEE Int'l. Conf. on Evolutionary Computation. Anchorage: IEEE, 1998. 834~839.
- [15] Brucker P, Drexel A, Möhring R, Neumann K, Pesch E. Resource-Constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research, 1999,112(1):3~41.
- [16] Xing WX, Xie JX. Advanced Computational Methods for Optimization. Beijing: Tsinghua University Press, 1999. 140~181 (in Chinese).

附中文参考文献:

- [16] 邢文训,谢金星.现代优化计算方法.北京:清华大学出版社,1999.140~181.

2004 年全国理论计算机科学学术年会

征 文 通 知

由中国计算机学会理论计算机科学专业委员会主办,海军工程大学信息与电气学院承办的“2004 年全国理论计算机科学学术年会”将于 2004 年 10 月在武汉召开。会议录用论文将收录在正式出版的论文集中,欢迎大家积极投稿。现将有关征文要求通知如下:

1. 应征论文应未在其他刊物或学术会议上正式发表过。特别欢迎有创见的论文和有应用前景的论文。

2. 稿件要求用计算机打印,格式为 38 行×38 字,字体为 5 号宋体。稿件中的图形要求画得工整、清晰、紧凑,尺寸要尽量小;图中字体要求为六号宋体。稿件正文不超过六千字。标题、作者姓名、作者单位、摘要、关键词采用中英文间隔行文。稿件各部分依次为:一、引言;二、...;最后是结束语。附录放在参考文献之后;参考文献限已公开发表的,文中最好不要出现文献序号。参考文献的格式为:

序号 作者·书名·出版社所在地:出版社名,出版年代

序号 作者·论文名·出处,年代·卷号(期号):起迄页码

务必附上第一作者简历(姓名、性别、出生年月、职称、学位、研究方向等)、通信地址和联系电话。并注明论文所属领域。请提供打印稿和电子稿各一份。来稿一律不退,请自留底稿。

3. 征文范围

程序理论(程序逻辑、程序正确性验证、形式开发方法等);计算理论(算法设计与分析、复杂性理论、可计算性理论等);语言理论(形式语言理论、自动机理论、形式语义学、计算语言学等);人工智能(知识工程、机器学习、模式识别、机器人等);逻辑基础(数理逻辑、多值逻辑、模糊逻辑、模态逻辑、直觉主义逻辑、组合逻辑等);数据理论(演绎数据库、关系数据库、面向对象数据库等);计算机数学(符号计算、数学定理证明、计算几何等);并行算法(分布式并行算法、大规模并行算法、演化算法等)。

4. 征文截止日期:2004 年 5 月 1 日

5. 论文投寄地址:(430033)武汉 海军工程大学信息与电气学院 张志祥 收

联系电话:027-83443985,83443984(张志祥,贲可荣)

电子信箱:tcs2004@vip.sina.com;hgzzx@163.com