

基于免疫算法多目标约束 P2P 任务调度策略研究

孟宪福,解文利

(大连理工大学计算机科学与技术学院,辽宁大连 116024)

摘要: 任务调度是 P2P 计算中的一项关键技术,直接影响到整个系统的计算性能.提出了基于免疫算法的多目标约束 P2P 任务调度策略.首先对多目标 P2P 任务调度相关问题做出定义,然后分别构造了考虑负载均衡的种群初始化算子和基于熵的克隆选择算子,并设计了新颖的交叉算子、变异算子和具有先验知识的疫苗.在描述了 P2P 节点获取和管理策略的基础上,提出了多目标任务调度策略.实验结果验证了调度策略在缩短任务执行时间和通信时间、以及节省调度费用等方面的有效性.

关键词: P2P; 任务调度; 免疫算法; 多目标约束

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2011) 01-0101-07

Research on P2P Task Scheduling with Multi-objective Constraints Based on Immune Algorithm

MENG Xian-fu, XIE Wen-li

(School of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning 116024, China)

Abstract: Since the task scheduling algorithm directly affects the performance of the P2P computing, the task scheduling with multi-objective constraints is presented by using immune algorithm. The population initialization operator considering load balance, the clone selection operator controlled by entropy, the new crossover operator, mutation operator and vaccine with apriori knowledge are designed for task scheduling based on model definition. And then the multi-objective task scheduling strategy is proposed after describing the mechanism for searching and managing the available P2P nodes. Experimental results indicate the validity of the proposed scheduling strategy in shortening the execution time and communication time, as well as saving the scheduling costs.

Key words: P2P; task scheduling; immune algorithm; multi-objective constraints

1 引言

任务调度是 P2P (Peer-to-Peer) 计算中的关键技术,直接影响到整个系统的计算性能^[1].多目标调度问题的解是一个非劣最优解的集合^[2],实际多目标问题往往是从非劣最优解集中选择一个可行的最佳方案.多目标优化问题的传统解决方法有目标加权法^[3]、Min-Min 法^[4]等.这些方法具有时间复杂度低、便于实现等特点,但是存在对参数要求高、所得非劣解少等缺点.同传统方法比,遗传算法具有隐含并行性、对问题依赖性小等优点,在多目标优化领域具有优越性^[5],但具有个体多样性减少得快、早熟收敛等缺点,不利于决策者做出有效选择.

人工免疫系统是一个高度并行、自适应和自组织的分布式系统,具有很强的学习、识别、记忆以及群体多样性等特点^[6].免疫算法已在单目标任务调度^[7]、多目标

优化^[8]等方面得到了有效应用.Yoo 等^[9]最早将人工免疫系统思想引入到多目标优化中,设计了基于免疫思想的遗传算法.Freschi 等^[10]提出了向量人工免疫系统 (Vector Artificial Immune System, VAIS).基于克隆选择理论,Gong 等^[11]提出了非劣邻域免疫算法 (Nondominated Neighbor Immune Algorithm, NNIA).在多目标优化的免疫算法中,NNIA 算法是比较有代表性的算法.

本文利用人工免疫系统和基于群体进化^[12]的思想来研究 P2P 环境下任务调度问题,通过设计新颖的克隆选择算子、交叉算子和变异算子,并利用具有先验知识的疫苗接种,提出了基于免疫算法的多目标约束 P2P 任务调度策略,该调度策略与传统的多目标遗传算法相比具有收敛速度快以及可获得更高质量解等优点,与已有的免疫算法相比在某些方面具有更优的性能.

2 相关模型和定义

2.1 网络模型和任务模型

本文利用无结构的 P2P 网络 Gnutella^[13]. 假设每个节点对任务的执行是采用非抢占式的, 调度到网络中的任务已经被划分为子任务, 并用有向无环图(DAG)表示任务间的依赖关系. 被调度到节点(处理机)上的所有任务被排成一个列表, 排列顺序代表了各子任务的先后执行顺序. 用 $S = (L_1, L_2, \dots, L_m)$ 来表示一个调度 S , 其中 $L_i = (T_{i1}, T_{i2}, \dots, T_{ik})$ 表示分配到节点 P_i 上的 k 个子任务的顺序.

任务关系 DAG 图表示为 $G = (T, E)$. 其中, T 为子任务集, 一个子任务 T_i 就是 DAG 中的一个节点. E 是有向边集, 表示子任务的执行顺序. $\langle T_i, T_j \rangle \in E$, 表示子任务 T_i 没有完成前, 子任务 T_j 不能执行. T_i 是 T_j 的前驱, T_j 是 T_i 的后继.

对于 n 个子任务、 m 个节点(以后若不特别说明, 均指处理机)的任务调度, 是指把这 n 个任务调度到 m 个节点上按已知的优先关系顺序执行. 下面定义用于任务调度的四个矩阵:

矩阵 $D_{n \times m}$, 其元素 $d_{ij} = 1$ 表示任务 T_i 被分配到节点 P_j 上执行, 否则 $d_{ij} = 0$, ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$).

矩阵 $E_{n \times m}$, 其元素 e_{ij} 表示任务 T_i 在节点 P_j 上的执行时间. 这里假设每个任务在各节点上的执行时间和各节点的单位时间使用费用已知, 则总的执行时间为 $\sum_{j=1}^m \sum_{i=1}^n e_{ij} d_{ij}$, 总的费用为 $\sum_{j=1}^m \sum_{i=1}^n e_{ij} d_{ij} \text{cost}(P_j)$, $\text{cost}(P_j)$ 表示使用节点 P_j 的单位时间费用, ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$).

矩阵 $C_{n \times n}$, 其元素 c_{ij} 表示任务 T_i 与任务 T_j 之间的通信量. 若两个任务被分配到同一个节点上, 则通信时间忽略不计. 若不在同一个节点上, 假设任务 T_i 和 T_j 被分别分配到节点 P_r 和 P_p 上, 则通信时间为两任务间通信量与所在节点间带宽的比值, 总的通信时间为:

$\sum_{r=1, i=1, p=1}^m \sum_{j=1, i=1, p=1}^n \frac{c_{ij}}{\text{bandwidth}(P_r, P_p)} d_{ir} d_{jp}$, $\text{bandwidth}(P_r, P_p)$ 表示节点 P_r 和节点 P_p 之间的带宽, 其中 ($i, j = 1, 2, \dots, n; r, p = 1, 2, \dots, m$).

矩阵 $R_{n \times n}$, 其元素 r_{ij} 表示任务间关系, $r_{ij} = 1$ 表示任务间可达, 否则 $r_{ij} = 0$, ($i, j = 1, 2, \dots, n$).

2.2 抗原表达和抗体表达

P2P 的特点决定了其任务调度性能是受多个因素制约的. 本文研究的 P2P 任务调度的目标就是优化任务的并行执行时间、任务之间的通信时间和调度费用.

若把抗原表达为各个目标函数, 则执行时间抗原

为 $Ag_1 = \sum_{j=1}^m \sum_{i=1}^n e_{ij} d_{ij}$, 通信时间抗原为 $Ag_2 = \sum_{r=1, i=1, p=1}^m \sum_{j=1}^n \frac{c_{ij}}{\text{bandwidth}(P_r, P_p)} d_{ir} d_{jp}$, 调度费用抗原为 $Ag_3 = \sum_{j=1}^m \sum_{i=1}^n e_{ij} \cdot d_{ij} \text{cost}(P_j)$, ($i, j = 1, 2, \dots, n; p, r = 1, 2, \dots, m$). 并令 $Ag_{1\xi}$ 、 $Ag_{2\xi}$ 、 $Ag_{3\xi}$ 分别为执行时间、通信时间和调度费用的阈值, 分别表示决策者对执行时间、通信时间和调度费用的最低要求.

抗体表达为调度问题的解. 本文利用文献[14]的高度定义来表达任务调度的解. 任务高度

$$H(T_i) = \begin{cases} 0 & \text{如果 } \text{Pred}(T_i) = \Phi \\ 1 + \max_{T_j \in \text{Pred}(T_i)} (H(T_j)) & \text{其他情况} \end{cases} \quad (1)$$

调整后的高度为

$$G(T_i) = \begin{cases} \text{Highest} & \text{如果 } \text{Succ}(T_i) = \Phi \\ \min_{T_k \in \text{Succ}(T_i)} (H(T_k)) - 1 & \text{其他情况} \end{cases} \quad (2)$$

其中, $\text{Highest} = \max(H(T_i))$, $\text{Pred}(T_i)$ 表示任务 T_i 的前驱任务集, $\text{Succ}(T_i)$ 表示任务 T_i 的后继任务集.

任务的实际高度 $HG(T_i)$ 为 $H(T_i)$ 与 $G(T_i)$ 之间的随机值. 若调度到节点上的任务按高度值递增有序, 即满足约束关系, 则认为是合法解(抗体).

染色体定义为一个节点上的任务列表 $L_i = (T_{i1}, T_{i2}, \dots, T_{ik})$. 基因为任务列表中的任务, 即 T_{ik} 为染色体 $L_i = (T_{i1}, T_{i2}, \dots, T_{ik})$ 的一个基因.

3 免疫算法表述

基于群体进化的典型算法是 VEGA(Vector Evaluated Genetic Algorithms)算法^[12]. 由于 VEGA 算法采用适应值比例分配方案, 向量解个体的总体适应值相当于各目标的加权和, 只是加权指数随每代种群的分布变化. 这一结论已被 Richardson 等提出并得到了 Schaffer 本人的证实^[12].

根据群体进化思想和人工免疫系统知识, 本文设计的免疫算法流程示意图如图 1 所示.

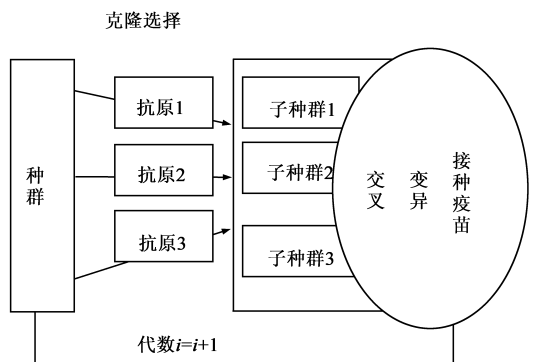


图1 免疫算法流程示意图

3.1 种群初始化

为保证调度的合法性和考虑负载均衡,首先根据高度值对任务进行划分,然后将可并行执行的任务尽量调度到不同节点上,并使各节点上的任务以高度值递增为序排成列表.种群规模由试验决定.对含有 m 个节点的网络进行任务调度,算法如下:

Step 1 计算每个任务的高度 $HG(T_i)$;

Step 2 分割任务,根据高度值 $HG(T_i)$,划分为 $Highest + 1$ 个高度集 $SG(h_i)$, $SG(h_i)$ 中的任务高度都为 h_i , $SG(h_i)$ 内的任务数目为 $NG(h_i)$;

Step 3 按 h_i 的升序对每个 $SG(h_i)$, 执行 Step 4;

Step 4 若 $NG(h_i) < m$, 随机选 $NG(h_i)$ 个节点, 为各节点分配一个任务, 转 Step 7; 否则转 Step 5;

Step 5 循环 $(m - 1)$ 次为前 $(m - 1)$ 节点分配任务;

由于 $SG(h_i)$ 中的所有任务是可以并行执行的, 把它们调度到不同的节点上能够提高整个任务的并行性, 既能使节点负载均衡, 又能获得更短的完成时间. 因此, 为 P_i 节点分配 q_i 个任务.

$$q_i = \lfloor NG(h_i) / m \rfloor + \left\lfloor \frac{\text{random}(0, 2 \times \lfloor NG(h_i) / m \rfloor + 1) - \lfloor NG(h_i) / m \rfloor}{2} \right\rfloor$$

其中 $\lfloor x \rfloor$ 为取不大于 x 的最大整数.

Step 6 将剩余的任务分配给最后一个节点;

Step 7 所有 $SG(h_i)$ 的任务分配完后, 对各节点上的任务按高度值递增排序, 并为 $D_n \times m$ 赋初值.

3.2 克隆选择算子

克隆选择算法的基本思想来源于免疫系统, 结构上经过“初始化种群; 评价标准计算; 种群间个体信息交互; 新种群产生”的过程. 算法描述如下:

Step 1 初始化种群, $NPOP$ 为种群大小;

Step 2 计算抗体-抗原亲和力 $IAG_{ij}(k)$; $IAG_{ij}(k)$ 为第 k 代种群中第 j 个抗体 $Ig_j(k)$ 与抗原 Ag_i 之间的适应函数值, ($i = 1, 2, 3; j = 1, 2, 3, \dots, NPOP$).

Step 3 根据亲和力、克隆规模及多目标选择熵, 对抗体进行克隆选择操作, 使每一个抗原产生一个抗体子种群, 并保留各子种群最优个体值;

由信息熵理论^[15]可知, n 个状态概率越接近, 则系统的信息熵越大. 为维持抗体群的数目不变及避免抗体过分集中于各目标接近的解集, 定义抗体 $Ig_j(k)$ 的多目标选择熵为

$$e(Ig_j(k)) = \sum_{i=1}^3 (-Ps(Ig_j(k)) * \ln Ps(Ig_j(k))).$$

其中, $Ps(Ig_j(k))$ 为抗体 $Ig_j(k)$ 的克隆选择概率, 是以亲和力比例自适应变化的, $Ps(Ig_j(k)) = \frac{IAG_{ij}(k)}{NSUM_i(k)}$,

$$NSUM_i(k) = \sum_{NPOP} IAG_{ij}(k).$$

根据多目标选择熵的定义, 抗体 $Ig_j(k)$ 对各目标贡献程度越大时, 抗体 $Ig_j(k)$ 的多目标选择熵越大. 由于所有抗体的多目标选择熵之和不一定为 1, 所以在用抗体的多目标选择熵进行克隆选择时, 将多目标选择熵调整为 $E(Ig_j(k)) = \frac{e(Ig_j(k))}{\sum_{j=1}^{NPOP} e(Ig_j(k))}$. 根据 $E(Ig_j(k))$

选择各子种群数目 80% 的抗体保留到下一轮免疫操作, 另外的 20% 由新产生的抗体补充, 其中, 子种群规模为 $\frac{NPOP}{3}$.

Step 4 计算新的抗体-抗原亲和力, 若有个体值优于最优个体值, 则更新最优个体值.

3.3 交叉算子

交叉算子通常作用在两个不同的抗体上, 把两个抗体染色体中的一部分基因相互交换, 产生两个新的抗体. 按交叉概率 P_c , 选择两个抗体 $Ig_i(k)$ 和 $Ig_j(k)$, 进行单点交叉, 算法如下:

Step 1 随机取一个整数 x , $0 \leq x \leq Highest$;

Step 2 在 $Ig_i(k)$ 和 $Ig_j(k)$ 的所有染色体 L_i ($1 \leq i \leq m$) 中选择交叉点; 若紧邻交叉点的任务为 T_i 和 T_j , 交叉点须满足条件 $HG(T_i) \leq x, HG(T_j) > x$;

Step 3 对 $Ig_i(k)$ 和 $Ig_j(k)$ 的各染色体分别交换相对应交叉点右部, 生成新抗体 $Ig_i(k)$ 和 $Ig_j(k)$;

为消除因基因缺失或重复造成的不合法解, 在交叉重组前进行如下操作: 设抗体 $Ig_i(k)$ 中在交叉点前的任务和交叉点后的任务分别组成两个集合 Set_{i1} 和 Set_{i2} , 抗体 $Ig_j(k)$ 所对应的为 Set_{j1} 和 Set_{j2} . 比较 Set_{i1} 和 Set_{j1} , 可得两个差集 $M1 = Set_{i1} - Set_{j1}$ 和 $M2 = Set_{j1} - Set_{i1}$, 令 $M = M1 \cup M2$, (比较 Set_{i2} 和 Set_{j2} , 可得与 M 集结果相同的集合).

Step 4 如果 $M = \Phi$, 结束; 否则对新的抗体 $Ig_i(k)$ 和 $Ig_j(k)$ 中所有染色体, 删除交叉点前的任务 T_k ($T_k \in M$), 然后按高度值递增顺序将 T_k 插入与任务 T_k 所在染色体的杂交对象中.

3.4 变异算子

变异算子用于将个体中的某个基因位以一定的概率进行改变, 以保证算法搜索到整个空间中的每一个个体. 按变异概率 P_m 分别对抗体群进行高频变异, 其算法如下:

Step 1 选择一个任务 T_i ;

Step 2 选择一个与 T_i 具有相同高度或通过 $R_n \times n$ 矩阵选择一个不可达任务 T_j ;

Step 3 交换 T_i, T_j , 形成新抗体. 因为具有相同高度的或不可达关系的任务是可并行执行的.

3.5 接种疫苗

接种疫苗是按问题的先验知识来修改抗体染色体上的某些基因,使某个抗体以较大概率具有更高的适应值.疫苗库是由与各目标对应的子疫苗库组成,其中,执行时间子疫苗库为 $\{\min_{j=1}^m e_{ij} \mid i=1,2,\dots,n\}$,即各任务在节点上的最小执行时间;通信时间子疫苗库为 $\{\max_{j=1}^n c_{ij} \mid i=1,2,\dots,n\}$,即各任务与其它任务间的最大通信量;调度费用子疫苗库为 $\{\min_{j=1}^m e_{ij} \text{cost}(P_j) \mid i=1,2,\dots,n\}$,即各任务在节点上的最小调度费用.疫苗接种算法如下:

Step 1 随机取一个整数 $x, 1 \leq x \leq \text{Highest}$;

Step 2 计算抗体的每一条染色体中高度值为 x 的数目,得到数目最大的染色体记为 L_i ;

Step 3 从数目最大的染色体 L_i 中选择一个高度值为 x 的任务 T_i .

Step 3.1 进行执行时间疫苗 $\min_{j=1}^m e_{ij}$ 的接种,若 $d_{ij} = 0$,把疫苗接种到染色体 L_j 上,即把子任务调度到执行时间最小的节点上,否则不做接种;

Step 3.2 进行通信时间疫苗 $\max_{j=1}^n c_{ij}$ 的接种,假设 T_j 被调度到节点 P_p 上,若 $i \neq p$,把疫苗接种到染色体 L_p 上,即把具有较大通信量的任务 T_i 和任务 T_j 调度到同一个节点上,否则不做接种;

Step 3.3 进行调度费用疫苗 $\min_{j=1}^m e_{ij} \text{cost}(P_j)$ 的接种,若 $d_{ij} = 0$,把疫苗接种到染色体 L_j 上,即把任务调度到调度费用最小的节点上,否则不做接种.

3.6 完全算法

Step 1 输入目标函数作为抗原的侵入;

Step 2 $k = 0$,初始化种群,设定算法参数;

Step 3 计算抗体与各抗原的亲合力 $IAG_{ij}(k)$, $\{IAG_{ij}(k)\} = \{IAG_{i1}(k), IAG_{i2}(k), \dots, IAG_{iNPOP}(k)\}$, ($i = 1, 2, 3$);

Step 4 依据克隆选择算子,对种群进行克隆选择,生成各子种群, $\{Ig_{ij}(k)\} = \{Ig_{i1}(k), Ig_{i2}(k), \dots, Ig_{iNPOP/3}(k)\}$, ($i = 1, 2, 3$);

Step 5 若满足终止条件终止计算,否则继续;

对各子种群抗体适应值排序并取前 20% 的抗体作为有益解集,算法的终止条件为有益解集内各抗体适应值均满足阈值 $Ag_{1\xi}$ 、 $Ag_{2\xi}$ 、 $Ag_{3\xi}$ 或 k 达到指定代数.假设第 t 代终止计算,则所得解为 $\{Ig_{ij}(t)\} = \{Ig_{i1}(t), Ig_{i2}(t), \dots, Ig_{iNPOP/3 \times 20\%}(t)\}$, ($i = 1, 2, 3$), 其中 $Ig_{11}(t)$, $Ig_{21}(t)$, $Ig_{31}(t)$ 为各目标的最优值.

Step 6 混合各子种群,生成临时种群 (c) ,

$\{(c)\} = \{Ig_{i1}(k), \dots, Ig_{iNPOP/3}(k)\}$, ($i = 1, 2, 3$);

Step 7 对种群 (c) 进行交叉、变异和接种疫苗;

依据交叉算子、变异算子和接种疫苗算子对临时种群进行交叉重组、高频变异和疫苗接种.若所得抗体与相应抗原的亲合力更好,则用新抗体更新旧的抗体,否则抗体保持不变.

Step 8 $k = k + 1$, 转 Step 3.

对 n 个子任务、 m 个节点、目标数为 3、抗体群规模为 $NPOP$ 、有益解集规模为 $NPOP \times 20\%$ 、各抗体染色体数目为 m 、最大迭代次数为 k 时,时间复杂度最差为: $O(NPOP \log n + kNPOP + k \log n + km)$. NNIA 算法通过拥挤距离比较过程减少非劣个体的数目,使最大的时间复杂度为 $O(NPOP \log NPOP)$, 本文算法复杂度与 NNIA 是同一数量级的.

4 任务调度

4.1 节点获取

由于 P2P 网络节点的动态不确定性,在任务调度之前需采用改进的 Gnutella 消息发送机制来定位调度节点.任务调度节点通过发送 Query 消息,在 TTL 范围内来获取满足调度条件的节点.接收到请求的节点根据任务的需求和节点自身的性能做出是否响应的判断,并生成一个 QueryHit 消息(对 Query 消息的回复),按照 Query 消息原路径发送给任务调度节点.改进的 QueryHit 消息头如图 2 所示.

MsgID	PID	Online	TTL	Hop	Performparam
-------	-----	--------	-----	-----	--------------

图 2 QueryHit 消息结构

其中,MsgID 为消息标识符;PID 标识消息的发送者,在 TTL 内,它是指最后一个转发消息的节点;Online 用于标识节点的空闲时间段;TTL 为生命周期;Hop 为路由跳数;Performparam 为获取的节点参数,主要包含该节点的 IP 地址、单位时间使用费用等.任务调度节点接收到应答信息后,选择一定数量满足条件的空闲节点进行调度.

4.2 节点动态性管理

节点的动态性管理是研究 P2P 环境下任务调度策略必须要考虑的重要问题.基于本文算法的特点,当有被调度节点(例如节点 P_k)退出网络时,调度节点将采用下述处理策略:

Step 1 更新执行时间和调度费用子疫苗库;

对各退出节点,检查执行时间子疫苗库 $\{\min_{j=1}^m e_{ij} \mid i = 1, 2, \dots, n\}$ 和调度费用子疫苗库 $\{\min_{j=1}^m e_{ij} \text{cost}(P_j) \mid i = 1, 2, \dots, n\}$, 若 $k = j$, 则更新相应疫苗(由于通信时间疫苗与节点无关,通信时间疫苗不做更新), 否则不做更新.

Step 2 对每一个未完成任务 $T_i \in P_k$, 若存在空闲的被调度节点, 则将任务 T_i 按高度值递增顺序调度到该节点上, 否则比较其它被调度节点上高度集为 $HG(T_i)$ 中的任务数 $NG(h_i)$, 若各 $NG(h_i)$ 值相同, 则将任务 T_i 按高度值递增顺序插入到执行时间最短的节点的任务列表中, 否则将任务 T_i 按高度值递增顺序插入到 $NG(h_i)$ 值最小的节点的任务列表中。

4.3 调度完全算法

Step 1 任务提交; 提交描述任务的脚本, 包括任务本身的属性(DAG形式)、子任务约束条件和静态资源的需求, 并为 $R_{n \times n}$ 矩阵和 $C_{n \times n}$ 矩阵赋值。

Step 2 节点获取; 根据 4.1 节获取策略, 在 TTL 内查找满足调度条件的节点, 并为 $E_{n \times m}$ 矩阵赋值。

Step 3 任务调度; 根据 3.6 节算法所得抗体群, 决策者根据需要来选择满意的调度策略并更新 $D_{n \times m}$ 矩阵。比如, 希望执行时间短时, 选择抗体 $I_{g_{11}}(t)$ 所表达的调度方式, 将任务调度到所选节点上运行。

Step 4 任务运行时管理; 在任务运行过程中, 调度节点对每个被调度节点的运行状态进行监控, 如果有节点退出, 则利用 4.2 节的节点管理策略对分配给该节点上的未完成任务进行重新调度。

5 仿真实验

5.1 实验参数选取

本文的仿真实验是利用基于 NS-2.27 和 PDNS 开发的 GnutellaSim 模拟器^[16]实现的。生成 500 个节点, 随机选择一定数量的含有所需资源节点。所用 DAG 图随机生成, 每个任务节点的后继为 [1, 3] 之间的随机数, 运行时间为 [1, 50] 之间的随机数, 子任务之间的传输延时是 [1, 20] 之间的随机数, 每个结点的单位时间使用费用是 [1, 10] 之间的随机数。算法主要参数为: 生命周期 $TTL = 2$, 种群大小 $NPOP = 120$, 终止代数 $k = 200$, 交叉概率 $P_c = 0.8$, 变异概率 $P_m = 0.2$, 阈值 $Ag_{1\zeta}$ 、 $Ag_{2\zeta}$ 、 $Ag_{3\zeta}$ 各组实验取值见表 1。

表 1 阈值 $Ag_{1\zeta}$ 、 $Ag_{2\zeta}$ 、 $Ag_{3\zeta}$ 的取值

节点个数	子任务数	$Ag_{1\zeta}$ 值	$Ag_{2\zeta}$ 值	$Ag_{3\zeta}$ 值
10	50	351	138	1770
20	50	256	119	1320
10	100	917	362	4400
20	100	669	307	3350
10	150	1550	618	6950
20	150	1265	520	6470

5.2 调度结果与分析

非劣分层遗传算法(Nondominated Sorting Genetic Algorithm, NSGA)^[17]采用优序排队选择方法强调好个体, 共享技术维持优秀个体。文献[18]在 NSGA 基础上提出

NSGA-II 算法, 算法以较快速度分层构造非支配集, 因在遗传算法中有较好时间效率和收敛性, 常被用于同类算法的比较对象。

NNIA 算法^[11]模拟多样性抗体共生、少数抗体激活的现象, 通过基于非受支配邻域的个体选择方法, 并根据活性抗体的拥挤程度进行比例克隆, 采用有别于 GA 的重组操作和变异操作, 以加强对当前非劣解前端中较稀疏区域的搜索。NSGA-II 和 NNIA 算法都没有考虑节点退出的情况。

表 2 任务调度的执行时间比较

节点个数	子任务数	执行时间及标准差							
		本文算法		NSGA-II 算法		NNIA 算法		节点退出	
10	50	361	8.6	365	9.3	360	8.7	407	13.6
20	50	266	7.5	269	7.4	264	7.8	316	11.8
10	100	924	6.9	927	6.2	925	6.8	1095	12.3
20	100	675	5.4	703	6.7	678	6.0	801	13.2
10	150	1558	6.6	1596	7.0	1581	6.3	1766	10.5
20	150	1275	5.8	1325	5.9	1296	6.4	1484	16.2

表 3 任务调度的通信时间比较

节点个数	子任务数	通信时间及标准差							
		本文算法		NSGA-II 算法		NNIA 算法		节点退出	
10	50	145	7.6	147	8.6	143	8.0	167	15.2
20	50	127	5.4	132	5.3	126	6.1	186	10.5
10	100	370	6.1	379	4.2	374	5.7	435	8.9
20	100	310	2.8	355	3.2	324	4.6	401	6.2
10	150	623	4.6	641	4.4	630	4.4	878	13.4
20	150	530	8.5	564	10.2	535	10.1	794	15.8

表 4 任务调度的调度费用比较

节点个数	子任务数	调度费用及标准差							
		本文算法		NSGA-II 算法		NNIA 算法		节点退出	
10	50	1786	8.5	1804	10.4	1782	10.1	1829	20.5
20	50	1332	6.5	1363	11.0	1328	7.9	1432	21.1
10	100	4420	9.3	4686	8.6	4425	9.5	4779	14.4
20	100	3374	8.5	3400	9.2	3390	7.2	3486	12.5
10	150	6972	7.4	7221	9.2	6984	9.9	8003	14.5
20	150	6486	9.4	6571	11.2	6552	8.2	7194	16.0

表 5 任务调度的进化代数比较

节点个数	子任务数	进化代数及标准差							
		本文算法		NSGA-II 算法		NNIA 算法		节点退出	
10	50	78	1.2	88	1.4	76	1.3	100	2.8
20	50	77	0.9	86	1.2	77	1.0	92	1.8
10	100	94	1.1	101	1.4	95	1.3	122	2.5
20	100	92	0.7	98	0.9	94	0.9	111	1.7
10	150	124	1.3	136	1.2	126	1.4	147	2.7
20	150	118	1.3	125	1.5	121	1.1	133	2.4

对同一个 DAG 图, 分别用本文算法、NSGA-II 算法、

NNIA 算法以及用本文算法在节点退出比率为 10% 的情况下进行调度. 每组实验, 通过 10 次调度, 取所得解的平均值和平均进化代数, 并分别计算所得平均值与各计算结果的标准差. 执行时间、通信时间、调度费用和进化代数的实验结果分别见表 2、表 3、表 4 和表 5.

由实验结果可以看出, 本文算法和 NNIA 算法在执行时间、通信时间、调度费用和进化代数方面都优于 NSGA-II 算法, 说明免疫算法能够克服遗传算法早熟收敛的问题, 与遗传算法相比具有一定优势. 在任务数为 50 时, 本文算法所得实验结果的平均值相接近于 NNIA 算法所得实验结果的平均值, 但在任务数较多时, 本文算法所得实验结果明显优于 NNIA 算法, 说明本文算法在处理大规模调度问题时具有优势. 由各均值与实验结果之间的标准差可知本文算法标准差值总体上比 NSGA-II 算法和 NNIA 算法所得标准差值要小, 说明本文算法在保持解的均匀性上具有一定优势. 同时, 表中数据显示, 在有节点退出情况下, 本文算法所得实验结果与没有节点退出情况下所得实验结果相差不大, 说明本文提出的节点动态性管理策略是可行的和有效的, 其任务调度策略也是适用于 P2P 网络环境的.

6 结束语

P2P 环境下的多目标约束任务调度是 NP 完全问题. 本文通过利用人工免疫系统和基于群体进化的思想提出了用于解决多目标约束 P2P 任务调度的免疫算法和 P2P 节点管理策略. 通过优化抗体种群的初始化和设计相关免疫操作算子, 更好地保证了进化过程中解集向优的方向进化. 按适应值赋值方式, 多目标进化算法可分为基于 Pareto 最优概念与不基于 Pareto 最优概念两大类. 由于本文算法不是基于 Pareto 最优概念的算法, 今后将在构建基于 Pareto 最优概念的免疫算法以及在免疫算法相关参数的选取上进行深入研究.

参考文献:

- [1] D Doval, D O'Mahony. Overlay networks a scalable alternative for P2P[J]. IEEE Internet Computing, 2003, 7(4): 79 - 82.
- [2] 谢涛, 陈火旺, 康立山. 多目标优化的演化算法[J]. 计算机学报, 2003, 26(8): 997 - 1003.
Xie Tao, Chen Huowang, Kang Lishan. Evolutionary algorithms of multi-objective optimization problems[J]. Chinese Journal of Computers, 2003, 26(8): 997 - 1003.
- [3] Y Jin, M Olhofer, B Sendho. Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how? [A]. Proceedings of the Genetic and Evolutionary Computation Conference [C]. San Francisco: Morgan Kaufmann, 2001. 1042 - 1049.
- [4] T D Braun, H J Siegel, N Beck. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(6): 810 - 837.
- [5] 雷德明, 严新平, 吴智铭. 多目标混沌进化算法[J]. 电子学报, 2006, 34(6): 1142 - 1145.
Lei De-ming, Yan Xin-ping, Wu Zhi-ming. Multi-objective chaotic evolutionary algorithm [J]. Acta Electronica Sinica, 2006, 34(6): 1142 - 1145. (in Chinese)
- [6] 焦李成, 杜海峰. 人工免疫系统进展与展望[J]. 电子学报, 2003, 31(10): 1540 - 1548.
Jiao Li-cheng, Du Hai-feng. Development and prospect of the artificial immune system[J]. Acta Electronica Sinica, 2003, 31(10): 1540 - 1548. (in Chinese)
- [7] 徐震浩, 顾幸生. 用混合算法求解 Flow shop 调度问题[J]. 华东理工大学学报, 2004, 30(2): 234 - 238.
Xu Zhen-hao, Gu Xing-sheng. A hybrid method for scheduling problems of flow shop[J]. Journal of East China University of Science and Technology, 2004, 30(2): 234 - 238. (in Chinese)
- [8] 尚荣华, 焦李成, 马文萍, 等. 用于约束多目标优化的免疫记忆克隆算法[J]. 电子学报, 2009, 37(6): 1289 - 1294.
Shang Rong-hua, Jiao Li-cheng, Ma Wen-ping, et al. An immune memory clone algorithm for constrained multi-objective optimization[J]. Acta Electronica Sinica, 2009, 37(6): 1289 - 1294. (in Chinese)
- [9] J Yoo, P Hajela. Immune network simulations in multicriterion design[J]. Structural Optimization, 1999, 18(2 - 3): 85 - 94.
- [10] F Freschi, M Repetto. VAIS: an artificial immune network for multi-objective optimization [J]. Engineering Optimization, 2006, 38(8): 975 - 996.
- [11] M G Gong, L C Jiao, H F Du, et al. Multi-objective immune algorithm with nondominated neighbor-based selection [J]. Evolutionary Computation, 2008, 16(2): 225 - 255.
- [12] J D Schaffer. Multiple objective optimization with vector evaluated genetic algorithms [A]. Proceedings of the 1st International Conference on Genetic Algorithms and Their Application, Hillsdale, Mich [C]. USA: Lawrence Erlbaum Associates, 1985. 93 - 100.
- [13] S Saroiu, P K Gummadi, S D Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts [J]. Multimedia Systems, 2003, 9(2): 170 - 184.
- [14] E S H Hou. A genetic algorithm for multiprocessor scheduling [J]. IEEE Transactions on Parallel and Distributed Systems, 1994, 5(2): 113 - 120.
- [15] J L Jonathan, A Prugel-Bennett. Maximum entropy analysis of genetic algorithm operators [A]. Lecture Notes In Computer Science; Vol. 993 [C]. London, UK: Springer-Verlag, 1995. 14 - 24.
- [16] Q He, M Ammar, G Riley, et al. Mapping peer behavior to

packet-level details: a framework for packet-level simulation of peer-to-peer systems[A]. Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MAS-COTS)[C]. USA: IEEE Press, 2003. 71 – 78.

[17] N Srinivas, K Deb. Multiobjective optimization using nondom-

inated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3): 221 – 248.

[18] D Kalyanmoy, A Pratap, S Agrawal, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.

作者简介:



孟宪福 男, 1960年2月生于辽宁大连. 副教授. 研究方向为对等计算与网格计算等.



解文利 男, 1984年4月生于山东淄博. 硕士. 研究方向为对等网络环境下的任务调度.