

文章编号:1001-9081(2010)02-0479-03

## 基于免疫粒子群的嵌入式系统软硬件划分方法

邵岁锋, 张英杰

(湖南大学 计算机与通信学院, 长沙 410082)

(shaosufeng@163.com)

**摘要:**针对嵌入式系统软硬件划分问题,提出一种粒子群算法与免疫克隆选择算法相结合的免疫粒子群软硬件划分方法。该算法重新定义了亲和力、克隆算子、变异算子和选择算子,有效克服了粒子群算法容易陷入局部最优的缺点。仿真实验表明该算法有效提高了了解的精度,获得了更合理的软硬件划分结果。

**关键词:**软硬件;粒子群优化算法;免疫克隆;算子

**中图分类号:** TP302; TP181 **文献标志码:** A

## Hardware/software partitioning algorithm for embedded systems based on immune particle swarm optimization

SHAO Sui-feng, ZHANG Ying-jie

(School of Computer and Communication, Hunan University, Changsha Hunan 410082, China)

**Abstract:** Hardware/software partitioning is one of the most important issues of the design of embedded systems. This paper described an approach based on immune particle swarm optimization algorithm. Affinity, clone copy operator, clone hyper-mutation operator and clone selection operator were redefined. The experimental results show that the algorithm can not only escape from local extreme but also improve the precision of the optimal solution.

**Key words:** hardware/software; Particle Swarm Optimization (PSO); immune clone; operator

### 0 引言

软硬件划分是嵌入式系统协同设计的关键步骤,其基本任务就是:在满足某些约束条件下,将系统功能行为分配到一定软硬件系统结构上。其实质是一种多目标的组合优化问题<sup>[1]</sup>。

传统的软硬件划分采用手工进行,随着嵌入式系统结构变得越来越复杂,开发效率要求也更高,手工方式已经不能满足现实的需要,急需一种自动化的划分方法,近些年软硬件划分算法研究得到重视。软硬件划分是一种 NP 完全问题<sup>[2]</sup>,目前常见的划分方法有混合线性规划和启发式算法,如遗传算法、模拟退火、禁忌搜索、蚁群算法等<sup>[3]</sup>。这些算法在解决软硬件划分问题上都取得了一定的效果,但也存在着容易陷入局部最优、运算时间长等缺点。

微粒群算法是一种智能启发式算法,具有参数少、计算简单、收敛速度快、鲁棒性好等优点<sup>[3]</sup>。免疫算法(Immune Algorithms, IA)是一种借鉴生物免疫系统特性而形成的启发式搜索算法,具有保持解群多样性的特点<sup>[4]</sup>。文献[5]将微粒群算法引入到软硬件划分问题中提出了一种离散粒子群软硬件划分算法,比传统的遗传算法取得了更好的划分结果,但是并没有克服粒子群算法容易陷入局部最优的缺陷。本文针对这一缺陷引入了免疫克隆选择算法,通过克隆复制、克隆变异和克隆选择操作避免了算法陷入局部最优,提高了算法的全局收敛性以及解的精度。

### 1 软硬件划分问题描述

进行软硬件划分之前设计者应该进行体系结构定义。本文采用的目标系统结构包括一个微处理器  $p$ , 存储器、总线、

一个 ASIC  $H = \{h_1, h_2, \dots, h_n\}$ , 其中  $h_1, h_2, \dots, h_n$  是 ASIC 中具有不同功能的硬件电路,如图 1 所示。

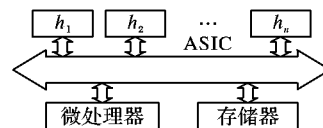


图1 系统结构

评价一个系统的性能要考虑软件执行的时间代价、内存消耗;采用硬件实现时的面积代价、功耗代价和开发成本等诸多因素。为了简化系统在本文中对于采用软件实现的功能模块只考虑其时间代价,对于使用硬件实现的功能模块考虑硬件面积和功耗两个方面的问题,不考虑各个功能模块之间的通信开销。假设嵌入式系统的设计说明已经采用合适的粒度大小,并转化为一个控制数据流程图(Control Data Flow Graph, CDFG)<sup>[5]</sup>。控制数据流程图作为软硬件划分问题的输入,每一个节点代表一个功能元件,它们可以用硬件或软件来实现。有向边代表各元件之间的通信和依赖关系。一个7个节点的简单控制数据流程图如图2所示。

此外每一个节点还包含了以下有用信息,采用软件实现的时候所需要的时间代价  $t(s)$ ;采用硬件实现的时候所需要的硬件面积  $h(s)$  和功耗信息  $p(s)$ 。评价一个划分结果的好坏,不仅要考虑实现的总代价  $Cost(s)$  还要考虑系统的总处理时间  $T(s)$ ,  $T(s)$  不能超过约束时间。控制数据流图中同层的节点可以并发执行,所以系统总的处理时间  $T(s)$  表示的是对控制数据流程图进行深度优先遍历时,各路径中耗时最多的路径所用的时间。 $Cost(s)$  包括硬件面积  $h(s)$  的总和  $H(s)$ , 功耗  $p(s)$  的总和  $P(s)$  两部分。

收稿日期:2009-08-07;修回日期:2009-09-15。

基金项目:国家自然科学基金资助项目(60634020);湖南省科技计划资助项目(2007GK3078)。

作者简介:邵岁锋(1983-),男,陕西凤翔人,硕士研究生,主要研究方向:智能控制;张英杰(1970-),男,湖南邵阳人,副教授,博士,主要研究方向:复杂工业过程计算机控制、智能控制、控制系统CAD。

整个划分算法就转化为一个规划问题:  $\min Cost(s)$ , 约束条件为  $T(s) < T_{\max}$ 。

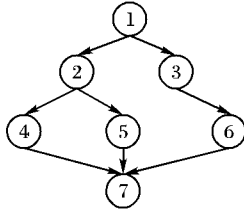


图2 控制数据流图(CDFG)

## 2 基于改进粒子群算法的划分方法

### 2.1 基本粒子群算法

PSO 算法是 Kennedy 和 Eberhart 根据鸟类的觅食行为提出的一种进化算法<sup>[7]</sup>。从诞生之日起在很多领域就得到了很好的应用,基本粒子群算法描述如下:

首先初始化一群随机粒子,然后每个粒子根据自身找到的最优解  $P_{best}$  和邻居粒子找到的全局最优解  $g_{best}$  来更新自身的位置,直到找到全局最优解。设  $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$  为微粒  $i$  的当前位置;  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  为微粒  $i$  的当前飞行速度;  $p_i = (p_{i1}, p_{i2}, \dots, p_{in})$  为微粒  $i$  所经历的适应值最大的位置,称为个体最好位置;所有微粒经过的最好位置为  $P_g(t)$ 。粒子  $x_i$  的位置和速度更新方程为:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(p_{ij} - x_{ij}(t)) + c_2r_{2j}(t)(p_{gj}(t) - x_{ij}(t)) \quad (1)$$

$$v_{ij} = \begin{cases} v_{\max}, & v_{ij} > v_{\max} \\ -v_{\max}, & v_{ij} < -v_{\max} \end{cases} \quad (2)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (3)$$

其中:  $w$  为惯性权重,表示先前粒子速度对当前速度的影响程度;  $x_{ij}$  表示第  $i$  个粒子在第  $t$  次迭代中位置的第  $j$  维分量;  $c_1$  和  $c_2$  为加速常数,通常在 0 和 2 间取值;  $r_1 \sim U(0,1)$ ,  $r_2 \sim U(0,1)$  为两个相互独立的随机函数。为了避免粒子飞出搜索空间,粒子速度通常被限制在  $[-v_{\max}, v_{\max}]$  中。

### 2.2 改进粒子群软硬件划分算法

算法实现的关键是要建立算法同软硬件划分问题之间的映射关系。由于基本粒子群优化算法是针对连续空间优化问题的求解而提出的,而软硬件划分的解空间是离散的,所以要对基本粒子群算法中粒子的位置、速度等进行重新定义。采用微粒群算法的离散二进制模型来实现<sup>[8]</sup>。

编码问题:我们采用二进制对 CDFG 中的节点进行编码,0 代表该节点采用软件实现,1 表示该节点使用硬件来实现。

粒子的位置  $X_i$ : 一个粒子代表一种潜在的划分方案,粒子每一个位置坐标  $x_{id}$  对应数据流图中的一个任务节点,  $x_{id}$  的取值代表节点的实现方式。例如:粒子  $i$  当前位置为  $X_i = \{0,1,0,0,\dots\}$ , 对应划分方案为 {软件实现,硬件实现,软件实现,软件实现, ...}。

粒子的速度  $V_i$ : 对粒子的速度  $v_{id}$  作 sigmoid 变换,使得  $v_{id} \sim U(0,1)$ 。粒子某一维的速度代表对应位极性发生翻转的概率。可以看出即使某一维的速度保持不变其对应位也可能根据速度的值发生极性的翻转。

粒子的位置更新方式:

$$x_{id} = \begin{cases} 1, & \text{rand}() < s(v_{id}) \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中:  $s(v) = \frac{1}{1 + e^{-v}}$ ;  $\text{rand}() \sim U(0,1)$  为一随机函数。

系统代价:

$$Cost(s) = P(s) + H(s) \quad (5)$$

约束条件:

$$T(s) \leq T_{\max} \quad (6)$$

其中:  $Cost(s)$  表示划分方案  $s$  所需要的系统总代价;  $P(s)$  表示硬件功能单元的总功耗代价;  $H(s)$  表示硬件功能单元的总面积代价;  $T(s)$  表示软件功能单元的总时间代价;  $T_{\max}$  代表系统容许的最大响应时间。

定义广义目标函数为:

$$\min Value(s) = a \times e^{\frac{T(s)-T_{\max}}{T_E}} \times |T(s) - T_{\max}| + b \times Cost(s) \quad (7)$$

该目标函数包括两部分信息。前半部分为惩罚函数,惩罚不满足时间约束的不可行解,绝对值项考虑到设计应该充分利用最大响应时间,而不是越快越好(越快意味着代价越高);后半部分为原目标函数的信息。参数  $a, b$  决定两部分所占的比重。  $T_E$  是控制惩罚力度的一个退火温度参数,算法每进行一次迭代则对  $T_E$  进行一次修正,本文取  $T_E = T_E \times 0.98$ 。随着算法进行惩罚力度越来越大,有利于加速算法收敛速度<sup>[9]</sup>。

适应值函数为:

$$Fitness(s) = 1/Value(s) \quad (8)$$

### 2.3 克隆选择算法

免疫算法是受到免疫系统启发而发展起来的一类仿生算法,该算法主要是模拟自然界生物免疫系统的机理和功能而实现<sup>[10]</sup>。克隆选择算法是免疫算法中基于克隆选择学说而提出。由于克隆选择机制中存在着克隆、超变异、抗体与抗原特异性结合、未被激发的细胞消亡及记忆细胞的产生等过程,因此,在保证收敛速度的同时又能维持解的多样性,可以提高解的精度避免陷入局部最优。

本文的做法是在粒子群算法外部建立一个规模为  $m$  的最优解集。每当粒子群算法出现一个新的全局最优解  $p_g$  就将其加入到最优解集中,最优解集中的解按照适应值大小递增有序排列。

#### 2.3.1 粒子亲和力的计算

抗体亲和力是种群中抗体优良性的标准。一般定义亲和力的大小与粒子的适应度成正比,与全局最优粒子的位置成反比,各个粒子亲和力之间的差异用定量的方式表现。本文把最优解集中的粒子看作抗体,采用粒子在最优解集中的排列序号这种定性的表示代替具体的定量表示,将粒子  $i$  的亲和力定义为:

$$affinity_i = p \quad (9)$$

其中  $p$  为粒子  $i$  在最优解集中的排列序号。

#### 2.3.2 克隆算子

根据亲和力的大小对最优解集中的每个粒子进行克隆操作。粒子  $i$  被克隆的数目为:

$$num_i = \left\lceil \frac{affinity_i}{\sum_{j=1}^m affinity_j} \times N \right\rceil \quad (10)$$

其中:  $\lceil \cdot \rceil$  表示向下取整;  $N$  为粒子群体的规模;  $m$  为最优解集规模,  $m$  取满足  $\sum_{j=1}^m j \leq N$  的最大正整数;克隆后的粒子总数

$Sum = \sum_{i=1}^m num_i$ , 很容易证明  $Sum \leq N$ 。在式(10)中由于  $affinity_i$  采用的是一种定性的描述方式,既体现粒子在最优解集中排位越靠后,也就是其适应值越大则被克隆的数目就越多;同时又由于定性的表示淡化了量之间的差距,避免了

与全局粒子最近的那些点克隆的数量过多从而容易陷入局部最优的缺点。

### 2.3.3 变异算子

目前常见的变异操作是根据克隆后个体的适应度来决定是否对粒子进行变异操作,对需要变异的粒子采用统一的方法进行变异,如高斯变异、柯西变异等。其缺点在于没有考虑需要变异的各粒子之间的差别。本文中对每个粒子都进行变异操作,其变异幅度跟亲和力大小相关。

最优解集中粒子 $i$ 被克隆后变成 $num_i$ 个具有相同亲和度的粒子,这些粒子变异位的个数相等,都按照以下定义得到:

$$d_i = \left\lceil \frac{m+1-p}{\sum_{j=1}^m j} \times N \right\rceil \quad (11)$$

其中: $\lceil \cdot \rceil$ 、 $p$ 、 $m$ 、 $N$  含义同式(10)。对这 $num_i$ 个粒子独立进行变异操作,每个粒子随机选择 $d_i$ 个位置进行变异操作。每个位置按位取反,即如果原位置为0则变为1,原位置为1则变为0。

从式(11)可以看出粒子亲和力越大变异位数越少,亲和力越小变异位数越多,这样既有利于保护最优解又有利于保持解的多样性。

### 2.3.4 选择算子

最优解集中的每个粒子经过克隆变异之后共产生 $sum$ 个新的粒子。如果 $sum = N$ ,则用这 $sum$ 个粒子替换当前群体粒子的位置;如果 $sum < N$ ,则用这 $sum$ 个粒子替换群体中 $sum$ 个粒子位置,其余 $N - sum$ 个粒子进行简单的移位变异。如选择移位步长为 $k$ ,则用粒子当前第1位的值去代替第 $k+1$ ,第2位的代替 $k+2$ ,依此类推,最后一位代替粒子的第 $k$ 位。这样一定程度可以保持粒子的多样性。

本文中直接用变异后的粒子来代替原粒子群操作简单方便,而且由于变异的位数跟亲和力相关,可以避免出现退化的情况。

## 3 免疫粒子群算法

算法流程如图3所示,具体实现步骤如下。

- 1) 在规定的搜索空间中随机初始化粒子群位置 $X_i$ ,速度 $V_i$ ,历史最优值 $p_i$ 和全局最优值 $p_g$ 。
- 2) 按照式(7)、(8)计算每个粒子的适应值,并根据适应值更新 $p_i$ 、 $p_g$ 以及外部最优解集 $m$ 。
- 3) 判断算法是否陷入局部最优。根据 $t$ 值是否超过给定阈值 $K$ 来判断算法是否陷入局部最优。 $t$ 用来累计全局最优解 $p_g$ 连续无更新的次数, $t$ 初始值为零,算法迭代完一次如果 $p_g$ 没有更新则 $t = t + 1$ ,如果 $p_g$ 有更新则将 $t$ 清零;如果 $t$ 没有达到阈值 $K$ 则继续向下执行,否则转6)。
- 4) 根据式(1)更新粒子的速度,根据式(4)更新粒子的位置。
- 5) 判断是否满足结束条件。满足则输出划分结果,否则转2)。
- 6) 免疫克隆选择操作:
  - ① 根据式(9)计算最优解集中每个粒子的亲和力;
  - ② 根据式(10)对最优解集中的粒子进行克隆复制操作;
  - ③ 根据式(11)对克隆的粒子进行高频变异操作;
  - ④ 根据选择算子更新每个粒子的当前位置。
- 转5)。

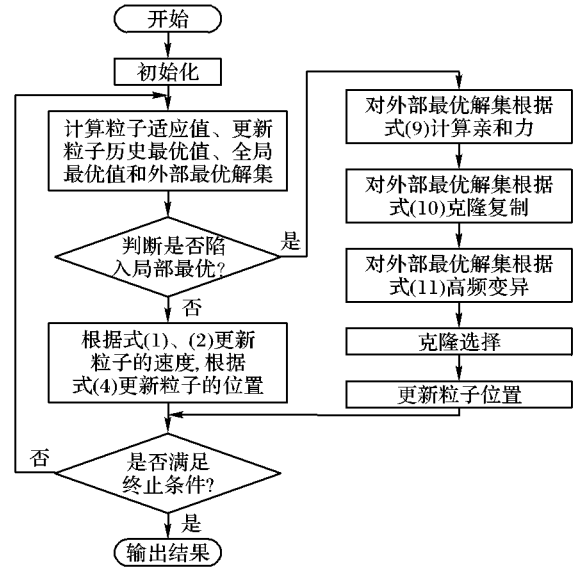


图3 算法流程

## 4 实验仿真及结果分析

首先随机生成10,20,40,60,100个节点的数据流图,同时随机生成各个节点的性能参数,并根据随机参数确定了约束条件以及约束条件的权值。数据流图的节点数目、软件代价、硬件代价、硬件功耗以及节点间的连接关系都是随机生成。

算法的参数设置如下。

种群规模 $N$ 取节点数的2倍。惯性权重:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter$$

其中: $w_{\max}$ 为0.9; $w_{\min}$ 为0.4; $iter_{\max}$ 为算法执行的最大迭代次数, $iter$ 为当前迭代次数;随着算法的迭代次数增加 $w$ 动态的减小,有利于加快算法收敛速度;学习因子 $c_1 = c_2 = 1.5$ ;速度 $v_{\max}$ 取10;算法每次迭代1000次,变异阈值 $K = 50$ 。

在同一PC机上使用本文的算法和文献[5]中的Discrete-PSO算法,使用相同的参数,分别对5种随机生成的控制数据流图进行了软硬件划分。两种算法都独立运行50次得到的最优结果和平均结果如表1所示。两种算法对不同节点所取得的平均系统总代价如图4所示。

通过表1可以看出在满足系统时间约束的条件下,对于10个节点12条边的数据流图两种算法取得了相同的最优值1207,但是本文算法的平均值要明显优于文献[5]中的算法。随着节点数目的增加,无论是最优值还是平均值本文均优于文献[5]。本文方法加入了免疫克隆选择算法,明显提高了了解的精度;不足之处是本文方法的运行时间略大于文献[5]中的方法。

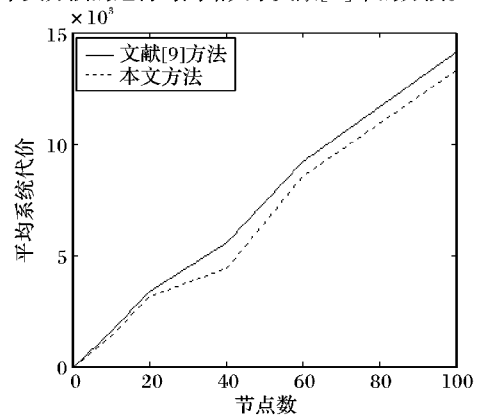


图4 平均系统代价比较

(下转第485页)

下面我们结合遗传算法进行求解。分别设定种群数量为50,100,200来进行求解。求解结果如表1,其中列出的迭代次数与求解时间均为得到最优解所需要的次数和时间。值得注意的是,考虑S1~S7时所需的求解时间比考虑S1~S5要少,这是因为遗传算法具有随机性,当问题规模较小时,求得最优解所需时间受随机性的影响较大。

从表1中可以看出,引入遗传算法有效地提高了求解效率,减少了求解时间。相对于用MILP模型直接求解,遗传算法的求解效率是其数十倍以上。

表1 不同参数下的迭代次数与求解时间

考虑对象	种群规模	迭代次数	求解时间/s
S1 ~ S5	50	9	17
	100	18	28
	200	41	74
S1 ~ S7	50	2	7
	100	8	23
	200	6	28
S1 ~ S10	50	50	210
	100	177	755
	200	207	1098

## 5 结语

本文针对机场地面滑行调度问题,提出了一种基于遗传算法的优化调度策略。该策略是针对文献[5]中提出的MILP模型进行的改进,能有效地提高求解效率,减少求解时间。原MILP模型中的变量分为两类,分别是决定各航班经过各交叉点顺序的变量 $Z_{ju}$ 和决定各航班经过各交叉点具体时间的变量 $t_{ju}$ 。本文将该MILP模型进行分解,利用遗传算法来求解 $Z_{ju}$ ,用LP模型来求解 $t_{ju}$ ,并对得到的解进行评估。通过模拟实验,证实遗传算法的引入将显著提高了求解效率,减少了求解时间。

(上接第481页)

表1 本文与文献[9]算法比较

节点/边	时间 约束	方法	软件 代价	硬件 代价	最优 总代价	平均 总代价
10/20	800	文献[5]方法	575	632	1027	1595.1
		本文	575	632	1027	1415.6
20/26	1000	文献[5]方法	858	1765	2625	3397.5
		本文	982	1537	2516	3146.6
40/53	1500	文献[5]方法	1459	3042	4501	5575.9
		本文	1467	2280	3747	4449.7
60/81	1500	文献[5]方法	1473	6669	8142	9214.5
		本文	1496	6155	7651	8563.2
100/57	2000	文献[5]方法	1998	11565	13563	14149.0
		本文	1951	10165	12116	13313.0

## 5 结语

本文针对嵌入式系统软硬件划分问题,提出了一种新的免疫粒子群算法,并与文献[5]中的粒子群算法进行了比较。实验结果表明本算法具有更好的全局收敛性和计算精度。下一步将在算法的稳定性方面做进一步的研究,同时将把算法引入到可重构系统中,扩大算法的应用范围。

### 参考文献:

[1] 邹谊,庄镇泉,李斌.基于量子遗传算法的嵌入式系统软硬件划

### 参考文献:

- [1] NEUMAN F, ERZBERGER H. Analysis of sequencing and scheduling methods for arrival traffic [R]. Moffet Field, CA: Ames Research Center, 1990.
- [2] HESSELINK H H, BASJES N. Mantea departure sequencer: Increasing airport capacity by planning optimal sequences [EB/OL]. [2009-07-08]. [http://atm2003.eurocontrol.fr/past-seminars/2nd-seminar-orlando-fl-usa-december-1998/papers/paper\\_011](http://atm2003.eurocontrol.fr/past-seminars/2nd-seminar-orlando-fl-usa-december-1998/papers/paper_011).
- [3] BOLAT A. Models and a genetic algorithm for a static aircraft-gate assignment problem [J]. Journal of the Operations Research Society, 2001, 52(10): 1107-1120.
- [4] SMELTINK J W, SOOMER M J, de Waal P R, et al. An optimisation model for airport taxi scheduling [EB/OL]. [2009-05-08]. <http://www.math.vu.nl/~mjsoomer/taxischeduling.pdf>.
- [5] RATHINAM S, MONTROYA J, JUNG Y. An optimization model for reducing aircraft taxi times at the dallas fort worth international airport [EB/OL]. [2009-05-20]. [http://www.aviationsystemsdivision.arc.nasa.gov/publications/surface/Rathinam\\_ICAS2008.pdf](http://www.aviationsystemsdivision.arc.nasa.gov/publications/surface/Rathinam_ICAS2008.pdf).
- [6] ROLING P C, VISSER H G. Optimal airport surface traffic planning using mixed-integer linear programming [EB/OL]. [2009-06-08]. <http://www.hindawi.com/journals/ijae/2008/732828.html>.
- [7] PESIC B, DURAND N, ALLIOT J-M. Aircraft ground traffic optimisation using a genetic algorithm [EB/OL]. [2009-06-20]. <http://www.recherche.enac.fr/opti/papers/articles/gecco01.pdf>.
- [8] 刘兆明,葛宏伟,钱锋.基于遗传算法的机场调度优化算法[J].华东理工大学学报:自然科学版,2008,34(3):392-398.
- [9] The Mathworks. Matlab, the language of technical computing [EB/OL]. The Math Works, Inc. [2009-08-01]. <http://www.mathworks.com/products/matlab/>.
- [10] LOFBERG J. YALMIP: A toolbox for modeling and optimization in Matlab [EB/OL]. [2009-06-11]. <http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php?n=Main.HomePage>.

分算法[J].电路与系统学报,2004,9(5):1-7.

- [2] GAREY M R, JOHNSON D S. Computers and intractability: A guide the theory of NP-completeness[M]. W. H. Freeman & Co, 1990
- [3] 熊志辉,李思昆,陈吉华.遗传算法与蚂蚁算法动态融合的软硬件划分[J].软件学报,2005,16(4):503-512.
- [4] 李莉,李洪奇,谢少龙,等.基于克隆选择的免疫粒子群优化算法[J].计算机科学,2008,35(10):253-255.
- [5] FARMAHINI-FARAHANI A, KAMAL A, FAKHRAIE S M, et al. HW/SW partitioning using discrete particle swarm [C]// Proceedings of the 17th ACM Great Lakes Symposium on VLSI. New York: ACM Press, 2007: 359-364.
- [6] 郑赞,黄国勇.通过遗传算法进行系统级软硬件划分[J].计算机辅助设计与图形学学报,2002,14(8):731-734
- [7] KENNEDY J, EBERHART R C. Particle swarm algorithm [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. New York: IEEE Press, 1995, 4: 1942-1948.
- [8] 吴启沅,汪镭.智能微粒群算法研究及应用[M].南京:江苏教育出版社,2005:184-186.
- [9] 张鲁峰,李思昆,刘功杰.遗传算法在软硬件划分中的应用[J].国防科技大学学报,2002,24(2):64-68.
- [10] DASGUPTA D. Artificial immune system and their applications [M]. Berlin: Springer-Verlag, 1999.
- [11] 刘丽珏,蔡自兴.基于克隆选择的粒子群优化算法[J].小型微型计算机系统,2006,27(9):1708-1710.