

# 改进的分布估计算法求解软硬件划分问题

余 娟 贺昱曜 冯晓华

(西北工业大学航海学院 西安 710072)

**摘 要** 软硬件划分是软硬件协同设计中的关键步骤,为 NP 难问题。分布估计算法可以解难优化问题,具有很好的全局搜索能力,但存在局部搜索能力差、种群多样性易失问题。针对此问题,对分布估计算法进行改进,对精英解进行克隆选择以加强局部搜索能力,对概率模型进行修正以改善种群多样性损失问题。同时,针对划分问题提出一种不可行解的修复方法。将改进后的分布估计算法应用于软硬件划分问题,并与现有算法做比较,结果表明所提算法在不同的约束条件下均可获得更好的优化结果。

**关键词** 分布估计算法,软硬件划分,精英克隆,概率模型修正,不可行解修复

**中图分类号** TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.054

## Solving HW/SW Partitioning Problem by Improved Estimation of Distribution Algorithm

YU Juan HE Yu-yao FENG Xiao-hua

(College of Marine, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract** Hardware/software (HW/SW) partitioning is a crucial step in embedded system co-design, also an NP hard problem. Estimation of distribution algorithm is good in globe search, but poor in local search and is prone to premature convergence because of diversity loss. An improved estimation of distribution algorithm was proposed to solve HW/SW partitioning problem. The improved algorithm clones and searches the promising solutions to strengthen the local searching ability, and corrects the probability model to improve the diversity loss. A method of repairing infeasible solutions was also proposed. Simulation was carried out. And the comparisons with existing algorithm demonstrate the effectiveness of the improved estimation of distribution algorithm in solving HW/SW partitioning problem.

**Keywords** Estimation of distribution algorithm, Hardware/software partitioning, Dominance clone, Probability model correction, Infeasible solutions repair

## 1 引言

典型的嵌入式系统包括特定的硬件部分(FPGAs 或 ASICs)和可编程部分即处理器(如 DSPs 或 ASIPs)<sup>[1]</sup>。系统中许多功能模块既可以由硬件来完成,也可以由软件来实现,一般而言软件实现相对比较灵活和廉价,而硬件实现可提高系统速度,但成本比较高。设计这样的软硬件混合系统时,将面临软硬件划分问题。软硬件划分的任务是在满足设计约束的条件下,以指定的系统开销最优为目标将不同的功能模块映射到目标结构中。软硬件划分的结果对产品最终的性能特征有非常重要的影响<sup>[2]</sup>。

软硬件划分算法有两类:精确算法和启发式算法。由于软硬件划分问题为 NP 难,随着问题规模增大,精确算法将无法求解。对于大规模问题需要使用启发式算法求解,如蜂群算法<sup>[2]</sup>、遗传算法<sup>[3]</sup>、粒子群算法<sup>[4]</sup>、禁忌搜索算法<sup>[5]</sup>、神经网络算法<sup>[6]</sup>、构造的启发式算法<sup>[7]</sup>。这些算法根据具体问题构建一定的规则求解,可以解决大规模问题,但不保证得到最优解,并且存在各自缺点,如禁忌搜索过于依赖初始解且为串

行搜索过程;遗传算法收敛慢,局部搜索能力弱;粒子群算法易产生早熟收敛;神经网络算法效率低,易陷入局部最优;蜂群算法在接近全局最优解时,搜索速度变慢,种群多样性减少等。

分布估计算法(Estimation of Distribution Algorithm, EDA)作为一种新兴算法<sup>[8]</sup>,具有很好的全局搜索能力,可以解决高维问题和难优化问题,因此近年得到广泛应用<sup>[9-13]</sup>,但目前文献中未见 EDA 在软硬件划分问题中应用。本文针对分布估计算法的缺点进行改进并将其应用于软硬件划分问题,将结果分别与基本 EDA 及文献[7]算法对比,实验表明,该算法的改进是有效的,能成功解决软硬件划分问题。

## 2 目标架构与软硬件划分模型

目标架构采用主处理器+协处理器结构。主处理器代表软件子系统(SW),采用软件的方式完成某一功能;而协处理器代表硬件子系统(HW),采用专用的硬件电路完成某一功能。

已有的软硬件划分算法大多仅考虑执行时间及成本,但

到稿日期:2013-11-12 返修日期:2014-02-17 本文受国家自然科学基金(61271143,60871080)资助。

余 娟(1980—),女,博士生,讲师,主要研究方向为嵌入式系统及优化算法, E-mail: yujuan@snnu.edu.cn;贺昱曜(1956—),男,博士,教授,博士生导师,主要研究方向为非线性控制理论、功率电子技术等;冯晓华(1975—),女,博士生,讲师,主要研究方向为群智能化算法。

是功耗这一因素也至关重要,因为功耗过高会导致芯片过热而破坏电路;另一方面,依赖电池供电的移动计算设备的有效工作时间对消费者的购买决定有很大影响,要想保证设备的有效工作时间,就应将功耗限制在一定范围内。因此本文在约束条件中加入了功耗这一条件,以功耗和执行时间为约束条件、以硬件成本最小为目标进行优化,与文献[7]相同。

为了便于比较,使用和文献[7]相同的划分模型,如图1所示。

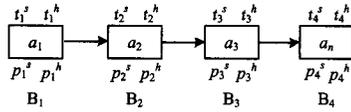


图1 划分模型

对于功能模块集合  $B = \{B_1, B_2, \dots, B_n\}$ , 软硬件划分问题要将集合中的所有功能模块映射至集合  $HW$  及  $SW$ , 且满足  $HW \in B, SW \in B, HW \cup SW = B, HW \cap SW = \Phi$ 。

根据划分算法的设计要求,对  $B_i$  进行了如下形式的参数化  $B_i = \langle a_i, t_i^s, t_i^h, p_i^s, p_i^h \rangle$ 。各参数含义: $a_i$  表示  $B_i$  采用硬件实现时的硬件成本,它与占用的电路面积成正比,软件成本可以用  $B_i$  需要的指令代码数和(或)需要的存储器数定义,软硬件划分算法中,通常可以不考虑该属性; $t_i^s, p_i^s$  为  $B_i$  用软件实现时的执行时间及功耗, $t_i^h, p_i^h$  为  $B_i$  用硬件实现时的执行时间及功耗。一般情况下,对于同一  $B_i$ ,用硬件实现时的功耗与执行时间都小于用软件实现<sup>[7]</sup>,所以设  $t_i^s > t_i^h, p_i^s > p_i^h$ 。

根据划分要求建立软硬件划分问题的数学模型如下:

$$\begin{cases} \min \text{cost}(X) = \sum_{i=1}^n a_i x_i \\ \text{s. t. } \sum_{i=1}^n [t_i^s(1-x_i) + t_i^h x_i] \leq T \\ \sum_{i=1}^n [p_i^s(1-x_i) + p_i^h x_i] \leq P \\ x_i \in \{0, 1\} \end{cases} \quad (1)$$

其中, $n$  表示功能模块个数, $x_i$  表示  $B_i$  的实现方式; $x_i = 1$  表示硬件实现, $x_i = 0$  表示软件实现; $T$  为时间约束, $P$  为功耗约束。

将式(1)整理变形为式(2):

$$\begin{cases} \min \text{cost}(X) = \sum_{i=1}^n a_i x_i \\ \text{s. t. } \sum_{i=1}^n t_i x_i \geq T' \\ \sum_{i=1}^n p_i x_i \geq P' \\ x_i \in \{0, 1\} \end{cases} \quad (2)$$

其中, $T' = \sum_{i=1}^n t_i^s - T, P' = \sum_{i=1}^n p_i^s - P, t_i = t_i^s - t_i^h, p_i = p_i^s - p_i^h$ 。本文采用改进的分布估计算法对式(2)表示的软硬件划分问题进行求解。

### 3 基本分布估计算法

分布估计算法是进化计算领域新兴的一个分支,是统计学习和进化算法的有机结合,是为解决遗传算法中存在的构造块破坏问题而提出的。EDA 通过对优势解集建立概率模型获得候选解的分布特征,然后对概率模型采样产生新的种群,如此反复进行,实现种群的进化。EDA 步骤如下:

Step 1 初始化种群;

Step 2 从种群中选择优势群体;

Step 3 由优势群体构建概率模型;

Step 4 随机采样;

Step 5 产生新的种群;

Step 6 判断终止条件是否满足,若是,则输出优化结果;否则,转 Step 2。

通过概率模型采样避免了对优势解的破坏。这与遗传操作中的交叉变异是完全相反的,也是区别于遗传算法的主要特征。分布估计算法从“宏观”上把握群体的进化方向,能更加有效地解决高维问题及难优化问题,降低时间复杂性。

然而,EDA 也存在缺点:1)局部搜索能力相对较差;2)进化过程中对概率模型学习时,容易对问题解空间的分布过拟合,若干代后不再产生多样性解从而出现早熟收敛。

### 4 改进的 EDA 求解软硬件划分问题

在大多数启发式算法中有一个根本的假设:优势解有着相似的结构<sup>[10]</sup>。这个假设对现实世界的问题是合理的。如旅行商问题中通过 Lin-Kernighan 方法获得的任意 2 个局部最优解的共同边平均值为 85%。

基于此理论,本文对种群中精英解集进行克隆选择,以加强原始算法的局部搜索能力,并对概率模型进行修正以缓解多样性损失,避免早熟收敛。采用改进后的 EDA 算法进行软硬件划分,将此算法记为 IEDA。

#### 4.1 解的表示

软硬件划分问题的解用长度为  $n$  的二进制字符串表示,即  $X = (x_1 \dots x_i \dots x_n)$ 。

#### 4.2 精英克隆选择

在基本 EDA 算法第 2 步之前加入精英克隆选择。

##### 4.2.1 精英个体克隆规模

在每一代群体  $pop(g)$  中,选择  $M_i$  个优势个体构成精英群体  $s^1(g)$  进行克隆,其中  $M_i = \lfloor \alpha * M \rfloor, M$  为种群规模,  $0 < \alpha < 1$ 。设  $nc_i$  为第  $i$  个精英个体  $X_i$  的克隆规模,确定  $nc_i$  时考虑两个因素:目标值  $\text{cost}(X_i)$  和  $X_i$  与种群中其余解的相似度,在本文划分问题中目标值  $\text{cost}(X_i)$  越小、 $X_i$  与其余解的相似度越小,说明  $X_i$  越具有优势,应加强对它的搜索。因此  $X_i$  的克隆规模  $nc_i$  通过如下公式计算:

$$nc_i = \left\lceil Nc * \frac{1/\text{cost}(X_i)}{\sum_{j=1}^{M_i} 1/\text{cost}(X_j)} * \frac{q_i}{\sum_{j=1}^{M_i} q_j} \right\rceil \quad (3)$$

其中, $Nc$  为克隆常数,是大于  $M$  的整数。 $q_i = \min\{d_{ij}\}, i = 1, 2, 3, \dots, M_s, j = i+1, \dots, M, d_{ij}$  为个体  $X_i$  与  $X_j$  之间的海明距离, $q_i$  越大说明  $X_i$  与种群中别的个体相似度越小。

##### 4.2.2 T 变换

设  $X_i$  的克隆体为  $X_i^j, j = 1, 2, \dots, nc_i$ , 对其进行如下操作  $T(X_i^j \rightarrow X_i^k)$ : 随机选取  $X_i^j$  中  $k$  个值为 0 的基因位变为 1,  $k+t$  个为 1 的基因位变为 0, 在软硬件划分问题中的物理意义是将  $k$  个软件域与  $k+t$  个硬件域中的功能模块交换映射域。

##### 4.2.3 优势替换

设  $\text{cost}(X_i^k) = \min(\text{cost}(X_i^j))$ , 其中  $X_i \in s^1(g), j = 1, 2, \dots, nc_i$ , 若  $\text{cost}(X_i^k) \leq \text{cost}(X_i)$ , 则  $X_i^k$  替代  $X_i$ ; 否则,  $X_i$  保持不变。

对于  $X_i \in s^1(g)$ , 精英克隆选择操作的过程是在  $X_i$  的邻域内, 提供了  $nc_i$  个不同的搜索方向, 在  $X_i$  的邻域内进行了

局部搜索,最后优势替换保证了所得解至少不会变差,并且加快了收敛速度。优势替换后形成新的种群  $pop'(g)$ 。

#### 4.3 概率模型

概率向量  $\rho=(\rho_1, \dots, \rho_k, \dots, \rho_n)$  为描述解空间分布的概率模型,  $\rho_k$  表示  $B_k$  映射至硬件域即  $x_k=1$  的概率。每一代 EDA 所产生的个体都通过对概率模型采样获得,即随机产生一个数  $r, r \in [0, 1]$ , 若  $r < \rho_i$ , 则  $x_i=1$ , 否则为  $x_i=0$ 。本文采用 UMDA<sup>[9,10]</sup> 概率模型。由  $pop'(g)$  中目标值较优的  $M_p$  个个体构成的优势群体  $s^2(g)$  构建概率模型, 其中  $M_p = \lfloor \beta * M \rfloor, 0 < \beta < 1$ 。  $\rho_i$  通过式(4)更新,  $x_i^j$  表示个体  $j$  的第  $i$  位取值:

$$\rho_i = \frac{\sum_{i \in s^2(g)} x_i^j}{M_p} \quad (4)$$

产生任意个体  $X$  的概率通过式(5)计算:

$$\rho(X) = \prod_{i=1}^n [\rho_i * x_i + (1 - \rho_i) * (1 - x_i)] \quad (5)$$

初始群体  $pop(0)$  在解空间按均匀分布随机产生, 即  $\rho=(0.5, 0.5, \dots, 0.5)$ 。

#### 4.4 概率模型修正

EDA 随着迭代的进行, 将会损失多样性, 种群方差越来越小, 最后衰减为零, 概率模型演化为只能产生相同结构解<sup>[14]</sup>。

EDA 多样性损失发生在 2 个阶段:

1. 通过概率模型采样产生种群时。这种多样性损失是由于采样产生的种群方差将小于父代种群方差造成的。4.2 节引入精英克隆选择不仅加强了局部搜索能力, 而且由于  $X_i^k$  是对采样产生的个体通过 T 变换而得, 因此可以改善种群多样性损失问题。

2. 通过对采样产生的  $M$  个个体选择以产生下一代的概率模型时。一般选择适应值高的个体构建概率模型, 虽然可得到优势解的分布特征, 但也会造成多样性损失, 这种情况可通过减缓概率模型的更新速度使之相对算法的搜索速度降低来缓解。一旦  $\rho_i$  足够接近 0 或 1, 种群中  $x_i$  的多样性就将接近 0。因此对概率模型进行边界修正, 如式(6)所示:

$$\rho_i' = \begin{cases} \gamma, & \rho_i < \gamma \\ p_i, & \lambda < \rho_i < 1 - \gamma \\ 1 - \gamma, & \rho_i > 1 - \gamma \end{cases} \quad (6)$$

其中,  $\rho_i$  为修正前概率,  $\rho_i'$  为修正后概率,  $\gamma = \frac{1}{n}$ 。

加入边界修正后, 阻止了  $\rho_i$  趋近 1 和 0, 降低了概率模型的更新速度, 也就改善了种群的多样性损失问题, 从而避免陷入局部最优。

#### 4.5 不可行解修复

采样获得的解不一定满足时间或功耗两个约束条件, 因此需要对不可行解进行修复。修复过程如下:

Step 1 计算  $tuse, puse; tuse = \sum_{i=1}^n t_i x_i, puse = \sum_{i=1}^n p_i x_i$ 。

Step 2 计算  $b_i$ :

若不满足时间约束即  $tuse < T' \&\& puse \geq P', b_i = a_i / t_i$ ;  
若不满足功耗约束即  $puse < P' \&\& tuse \geq T', b_i = a_i /$

$p_i$ ;

若时间、功耗约束均不满足即  $puse < P' \&\& tuse < T',$

$b_i = a_i / \sqrt{t_i^2 + p_i^2}$ 。

Step 3 将  $b_i$  按升序排序, 设排序后  $b_1 < b_2 < \dots < b_n$ 。

Step 4 for  $i=1:n$

若  $x_i=0$ , 则令  $x_i=1, tuse=tuse+t_i, puse=puse+p_i$ ;

若  $tuse \geq T' \&\& puse \geq P'$ , 结束循环。

#### 4.6 IEDA 算法步骤

输入:  $a_i, t_i^k, p_i^k, t_i^h, p_i^h, T, P$

输出:  $X_{bestsofar} = (x_1, x_2, \dots, x_n)$

Step 1 设置参数: 迭代次数  $G$ , 种群规模  $M$ , 克隆系数  $Nc$ , 优势群体系数  $\alpha, \beta, T$  变化参数  $k, t$ , 令  $g=0$ 。

Step 2 产生初始种群  $pop(0) = \{X_1, \dots, X_M\}$ , 修复不可行解, 计算目标函数值  $cost(X_i)$  并升序排列,  $cost_{bestsofar} = cost(X_1), X_{bestsofar} = X_1$ 。

Step 3 对  $pop(g)$  进行精英克隆选择操作, 形成新的种群  $pop'(g)$ 。

Step 4 根据  $pop'(g)$  的优势群体  $s^2(g)$  构建概率模型并对概率模型进行修正。

Step 5 采样产生新的种群  $pop(g+1)$  并修复不可解。

Step 6 计算  $cost(X_i)$  并升序排列,  $X_i \in pop(g+1)$ 。

Step 7 更新最优解。若  $cost(X_i) < cost_{bestsofar}$ , 则  $cost_{bestsofar} = cost(X_i), X_{bestsofar} = X_i$ 。

Step 8  $g=g+1$ 。

Step 9 判断是否满足终止条件, 若满足则输出结果  $X_{bestsofar}$ ; 否则转 Step 3。

#### 4.7 算法复杂度分析

算法由数个操作构成, 其中目标值计算的时间复杂度为  $O(M \cdot n)$ , 不可行解修复的时间复杂度最差为  $O(M \cdot n)$ , 排序算法时间复杂度为  $O(n \log n)$ , 优势克隆选择操作时间复杂度为  $O(n * \sum_{i=1}^{M_k} nc_i)$ , 概率模型的更新以及采样时间复杂度为  $O(M \cdot n)$ 。由于  $M, M_i$  为常数, 因此算法时间复杂度由排序算法决定, 为  $O(n \log n)$ , 即随着划分问题规模  $n$  的增长, 时间复杂度按线性对数增长。

### 5 实验

文献[7]对式(1)采用基于贪婪思想的启发式算法(HA)进行求解。本文使用 C 语言对基本 EDA、IEDA 以及 HA 算法在 36 种不同约束条件进行仿真比较。

#### 5.1 参数设置

##### 5.1.1 调度块参数

为了与 HA 算法进行比较, 参数设置与文献[7]相同,  $t_i$  在  $(0, 30)$  随机产生,  $p_i$  在  $(0, 20)$  随机产生;  $t_i^h, p_i^h$  分别在  $(0, \lambda * t_i), (0, \lambda * p_i)$  间随机产生;  $a_i$  在  $(0, 100)$  间随机产生。

##### 5.1.2 算法参数

根据经验, 种群规模  $M=40, Nc=25 * M, \beta=1/4; \alpha$  确定的是需要克隆的精英解数目,  $\alpha$  大可以进行更全面的搜索, 但也会带来更长的计算时间, 综合考虑, 令  $\alpha=1/8$ 。

在 5 种典型约束情况下对  $k$  取 1、2、3,  $t$  取 1、2 共 6 种组合进行仿真, 最终确定  $k=1, t=1$ 。

终止条件为达到最大迭代次数  $G, G=2000$ 。

#### 5.2 实验结果

设  $cost(*)$  为算法 \* 最终所得硬件成本,  $cost_{hw}$  为所有模块均映射至硬件域的硬件成本。令

$$cost\_used(*) = \frac{cost(*)}{cost_{hw}} \times 100\% \quad (7)$$

$$\begin{cases} \delta_{HA-EDA} = cost\_used(HA) - cost\_used(EDA) \\ \delta_{HA-IEDA} = cost\_used(HA) - cost\_used(IEDA) \end{cases} \quad (8)$$

由于本文优化目标是使成本最小,因此由式(7)可看出  $cost\_used(*)$  越小,说明该算法优化结果越好;由式(8)可看出若  $\delta > 0$ ,说明 EDA 或 IEDA 所获结果优于 HA 所获结果,且该值越大,说明越优于 HA;若  $\delta < 0$ ,则 EDA 或 IEDA 结果劣于 HA 结果。

由于对不同约束条件的软硬件划分问题进行求解,因此时间约束  $T(\theta_T)$  及功耗约束  $P(\theta_P)$  设定如式(9):

$$\begin{cases} T(\theta_T) = T_{hw} + \theta_T * (T_{sw} - T_{hw}) \\ P(\theta_P) = P_{hw} + \theta_P * (P_{sw} - P_{hw}) \end{cases} \quad (9)$$

$T_{hw}, P_{hw}$  为所有模块映射至硬件域所用时间、功耗; $T_{sw}, P_{sw}$  为所有模块映射至软件域所用的时间、功耗; $0 \leq \theta_T \leq 1, 0 \leq \theta_P \leq 1, \theta_T, \theta_P$  取不同的值代表时间约束及功耗约束的不同,可以看出,时间约束取值在  $[T_{hw}, T_{sw}]$  范围内,功耗约束取值在  $[P_{hw}, P_{sw}]$  范围内。

与文献[7]相同,  $n=100, \lambda=1/2$ , 在  $\theta_T, \theta_P$  分别取 1/6、2/6、3/6、4/6、5/6、6/6 共 36 种约束情况下对算法进行仿真,仿真结果如表 1 所列,结果均为 20 组随机数据计算结果的平均值。为了比较,也列出了  $\delta', \delta'$  为文献[7]使用 HA 算法所得解与最优解之间的误差,  $\delta'$  越小说明 HA 结果越接近最优解,其数据来自文献[7]。

表 1 仿真结果

		P(1/6)	P(2/6)	P(3/6)	P(4/6)	P(5/6)	P(6/6)
T(1/6)	cost-use(HA)	61.03	53.45	53.13	51.58	50.25	50.85
	cost-use(EDA)	60.93	52.97	52.79	51.50	50.38	50.89
	cost-use(IEDA)	60.06	52.36	52.47	51.32	50.08	50.75
	$\delta_{HA-EDA}$	0.1	0.48	0.34	0.08	-0.13	-0.04
	$\delta_{HA-IEDA}$	0.97	1.09	0.66	0.26	0.17	0.10
	$\delta'$	0.93	0.93	0.49	0.22	0.15	0.08
T(2/6)	cost-use(HA)	52.79	37.86	33.73	33.60	32.44	31.78
	cost-use(EDA)	53.55	38.29	33	33.20	32.40	32.02
	cost-use(IEDA)	52.00	37.09	32.32	32.81	32.04	31.47
	$\delta_{HA-EDA}$	-0.76	-0.43	0.73	0.40	0.04	-0.24
	$\delta_{HA-IEDA}$	0.79	0.77	1.41	0.79	0.40	0.31
	$\delta'$	0.77	0.84	1.41	0.75	0.39	0.22
T(3/6)	cost-use(HA)	52.97	33.83	22.67	19.93	18.86	18.28
	cost-use(EDA)	52.62	33.61	22.89	19.71	18.45	18.27
	cost-use(IEDA)	52.48	32.43	21.89	18.70	18.18	17.91
	$\delta_{HA-EDA}$	0.35	0.22	-0.22	0.22	0.41	0.01
	$\delta_{HA-IEDA}$	0.49	1.4	0.78	1.23	0.68	0.37
	$\delta'$	0.42	1.31	0.55	1.03	0.54	0.35
T(4/6)	cost-use(HA)	52.16	32.75	19.69	10.27	8.72	8.52
	cost-use(EDA)	51.96	32.50	18.98	10.23	8.34	8.09
	cost-use(IEDA)	51.89	31.98	18.75	9.87	8.15	8.08
	$\delta_{HA-EDA}$	0.20	0.25	0.71	0.04	0.38	0.43
	$\delta_{HA-IEDA}$	0.27	0.77	0.94	0.4	0.57	0.44
	$\delta'$	0.17	0.64	0.92	0.55	0.49	0.34
T(5/6)	cost-use(HA)	52.00	31.53	19.50	9.27	2.84	2.41
	cost-use(EDA)	51.89	31.51	19.20	9.02	2.85	2.31
	cost-use(IEDA)	51.83	31.11	18.90	8.58	2.71	2.26
	$\delta_{HA-EDA}$	0.11	0.02	0.30	0.25	-0.01	0.10
	$\delta_{HA-IEDA}$	0.17	0.42	0.60	0.69	0.13	0.15
	$\delta'$	0.16	0.40	0.52	0.47	0.12	0.15
T(6/6)	cost-use(HA)	50.41	32.15	17.79	8.64	2.54	0.01
	cost-use(EDA)	50.47	32.38	17.94	8.54	2.49	0
	cost-use(IEDA)	50.28	31.9	17.3	8.27	2.36	0
	$\delta_{HA-EDA}$	-0.06	-0.23	-0.15	0.10	0.05	0.01
	$\delta_{HA-IEDA}$	0.13	0.25	0.49	0.37	0.18	0.01
	$\delta'$	0.08	0.22	0.33	0.31	0.2	0

从表 1 所列数据可以看出:

1) 对比  $\delta_{HA-EDA}, \delta_{HA-IEDA}$  两行数据,除了(6/6,6/6)约束情况下基本 EDA 与 IEDA 均获得最优值 0 从而得到  $\delta$  相等外,其余 35 种情况均为  $\delta_{HA-IEDA} > \delta_{HA-EDA}$ ,说明 IEDA 算法相比基本 EDA 算法求解软硬件划分问题可以获得更好的优化结果,因此 IEDA 对基本 EDA 的改进是有效的。

2) 对于同一时间约束  $T(\theta_T), P(2/6), P(3/6), P(4/6)$  这 3 种情况下的  $\delta'$  值均大于  $P(5/6), P(6/6)$  两种情况下的  $\delta'$  值;对同一功耗约束  $P(\theta_P)$ ,存在同样情况,说明:①存在至少一个约束条件较松弛时;② $\theta_T, \theta_P$  相差较大时,即两个约束条件松弛程度相差较大时  $\delta'$  更小,HA 可以获得更好的结果。也就是说,文献[7]所使用的 HA 方法解决软硬件划分问题

结果的优劣是依赖于约束条件的。

3) 36 种约束情况下均有  $\delta_{HA-IEDA} > 0$ ,且  $\delta_{HA-IEDA}$  与  $\delta'$  变化趋势基本相同。 $\delta_{HA-IEDA} > 0$  说明 IEDA 优于 HA 所得结果; $\delta_{HA-IEDA}$  与  $\delta'$  分别为 HA 所获结果与 IEDA 所获结果以及最优解之差,二者变化趋势相同说明 IEDA 所获结果与最优解之间的误差基本为常数,即约束的大小不影响 IEDA 算法性能,无论是松弛约束还是严格约束,IEDA 算法都可获得好的划分结果。由于所得结果为随机产生的 20 组数据所获结果平均值,因此  $\delta_{HA-IEDA}$  与  $\delta'$  数值上存在微小误差,说明 IEDA 所获结果与最优值相差不大。

通过仿真结果可以看出,IEDA 算法可以更有效地解决软硬件划分问题。

**结束语** 分析了分布估计算法的优缺点,对基本的分布估计算法进行改进并将它用于求解软硬件划分问题。在分布估计算法中加入了精英克隆选择操作,加强了局部搜索能力,并对概率模型进行修正改善了多样性丧失的问题,改进后的算法进行软硬件划分问题的求解,以时间和功耗为约束条件、以成本为优化目标进行优化。将结果与基本 EDA 以及文献中算法相比,结果表明 IEDA 算法可以获得更优结果,可以有效求解软硬件划分问题。

## 参 考 文 献

- [1] Abdelhalim M B, Habib S E-D. An integrated high-level hardware/software partitioning methodology[J]. Design Automation for Embedded Systems, 2011, 15(1): 19-50
- [2] Koudil M, Benatchba K, Tarabet A, et al. Using artificial bees to solve partitioning and scheduling problems in codesign[J]. Applied Mathematics and Computation, 2007(186): 1710-1722
- [3] 邹谊, 庄镇泉, 杨俊安. 基于遗传算法的嵌入式系统软硬件划分算法[J]. 中国科学技术大学学报, 2004, 34(6): 724-731
- [4] Wu Yue, Zhang Hao, Yang Hong-bin. Research on parallel HW/SW partitioning based on hybrid PSO algorithm[J]. Algorithms and Architectures for Parallel Processing, 2009, 5574: 449-459
- [5] Wang P, Wu J G. Efficient heuristic and tabu search for hardware/software partitioning[J]. Computer, Science, 2012, 39(1): 290-294
- [6] Huang Yue, Kim Y. Applying hybrid neural fuzzy system to embedded system hardware/software partitioning[C]// Third International Conference on Intelligent Computing, ICIC 2007. 2007, 4682: 660-669
- [7] Wu Jigang, Srikanthan T. Algorithmic aspects of area-efficient hardware/software partitioning[J]. The Journal of Supercomputing, 2006, 38(3): 223-235
- [8] Larranaga P, Lozano J A. Estimation of distribution algorithms: A new tool for evolutionary computation[M]. Netherlands: Springer, 2002
- [9] Chen S-H, Chen M-C, Chang Pei-chann, et al. Guidelines for developing effective Estimation of distribution algorithms in solving single machine scheduling problems[J]. Expert Systems with Applications, 2010, 37(9): 6441-6451
- [10] Izquierdo C E, Velarde J L G, Melián-Batista B, et al. Hybrid Estimation of distribution algorithm for the quay crane scheduling problem[J]. Applied Soft Computing, 2013, 13: 4063-4076
- [11] Wang Ling, Wang Sheng-yao, Xu Ye. An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem[J]. Expert Systems with Applications, 2012, 39: 5593-5599
- [12] Ceberio J, Irurozki E, Mendiburu A, et al. A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems[J]. Progress in Artificial Intelligence, 2012, 1(1): 103-117
- [13] Santana R, Larrañaga P, Lozano J A. Combining variable neighborhood search and Estimation of Distribution Algorithms in the protein side chain Placement problem[J]. Journal of Heuristics, 2008, 14(5): 519-547
- [14] Branke J, Lode C, Shapiro J L. Addressing sampling errors and diversity loss in UMDA[C]// Proceedings of the 9th annual conference on Genetic and evolutionary computation. London, England, United Kingdom, 2007: 508-515
- (上接第 258 页)
- [2] SemEval2007[OL]. <http://nlp.cs.swarthmore.edu/semeval/>
- [3] Ghazi D, Inkpen D, Szpakowicz S. Hierarchical versus Flat Classification of Emotions in Text[C]// Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Los Angeles, California, 2010: 140-146
- [4] Bellegarda J R. Emotion Analysis Using Latent Affective Folding and Embedding[C]// Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Los Angeles, California, 2010: 1-9
- [5] Picard R W. Affective Computing[M]. Cambridge: MIT Press, 1997
- [6] Cosatto E, Ostermann J, Graf H P. Lifelike talking faces for interactive services[J]. Proc. IEEE, 2003, 91(9): 1406-1429
- [7] Ryan S, Scott B, Freeman H, et al. The Virtual University: The Internet and Resource-based Learning [M]. London, UK: Kogan, 2000
- [8] Abbasi A. Affect Intensity Analysis of Dark Web Forums[C]// Proc. IEEE Int. Conf. Intelligence and Security Informatics (ISI). New Brunswick, NJ, 2007: 282-288
- [9] Strapparava C, Mihalcea R. Learning to Identify Emotions in Text[C]// Proc. ACM Symposium on Applied computing. Fortaleza, Brazil, 2008: 1556-1560
- [10] Alm C. Affect in text and speech[M]. University of Illinois at Urbana-Champaign, Department of Linguistics, 2008
- [11] Wilson T, Wiebe J, Hoffmann P. Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis[J]. Computational Linguistics, 2009, 35(3): 399-433
- [12] Golder S A, Macy M W. Diurnal and Seasonal Mood Vary with Work, Sleep, and Day length Across Diverse Cultures [J]. Science, 2011, 333(6051): 1878-1881
- [13] Paltoglou G, Thelwall M. Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(4): 66
- [14] 谢丽星, 周明, 孙茂松. 基于层次结构的多策略中文微博情感分析和特征抽取[J]. 中文信息学报, 2012, 26(1): 73-83
- [15] 杨亮, 林原, 等. 基于情感分布的微博热点事件发现[J]. 中文信息学报, 2012, 26(1): 84-90
- [16] LIWC[OL]. <http://www.liwc.net/>
- [17] C-LIWC[OL]. <http://c-liwc.blogbus.com/>
- [18] HowNet[OL]. <http://www.keenage.com/>
- [19] 清华大学中文分词演示系统[OL]. <http://nlp.csai.tsinghua.edu.cn/app/wordSegment/>