

# 基于混沌 PSO 的动态可重构系统软硬件划分

张亮忠,熊选东,王松锋,付建丹  
(信息工程大学 电子技术学院,河南 郑州 450004)

**摘要:**软硬件划分是动态可重构系统软硬件协同设计中的关键技术之一,如何兼顾划分效率和划分效果,达到两者的最佳结合是软硬件划分的主要问题。在考虑动态部分重构及重构延时等特征的基础上,提出一种微粒群优化算法与混沌优化算法相结合的混沌微粒群软硬件划分方法。该算法使用基于实数编码的微粒群优化算法执行全局搜索,再根据搜索结果采用混沌优化算法执行局部搜索,具有较强的全局搜索和跳出局部最优的能力。仿真实验表明,该算法比标准微粒群算法和遗传算法具有更好的有效性和快速性,能够有效地实现应用任务图到可重构系统的时空映射。

**关键词:**可重构系统;软硬件划分;微粒群优化算法;混沌优化算法

**中图分类号:**TP302

**文献标识码:**A

**文章编号:**1673-629X(2012)06-0045-04

## Hardware/Software Partitioning Algorithm for Dynamically Reconfigurable Systems Based on Chaotic PSO

ZHANG Liang-zhong, XIONG Xuan-dong, WANG Song-feng, FU Jian-dan  
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

**Abstract:** Hardware/software partitioning is an important technology in hardware/software codesign of dynamically reconfigurable system. How to keep the balance between the speed and performance is the key problem. Considering the partially run-time reconfiguration and reconfiguration delay of reconfigurable system, a chaotic particle swarm optimization algorithm is proposed which combines the particle swarm optimization (PSO) and chaotic optimization algorithm (COA). This algorithm uses PSO based on real coding to search globally, and then uses COA to search locally, has strong ability of searching and getting out of the local optimum. The experimental results show that the method can efficiently map the task graph to reconfigurable system, and get better partitioning results and executing speed than the method based on PSO or genetic algorithm (GA).

**Key words:** reconfigurable system; hardware/software partitioning; particle swarm optimization; chaotic optimization algorithm

## 0 引言

可重构硬件将传统通用微处理器 (General Purpose Processor, GPP) 的可编程性与专用集成电路 (Application Specific Integrated Circuit, ASIC) 任务基于空间并行执行的特点结合起来,既能够提供近似于 ASIC 的执行效率,又拥有类似于 GPP 的灵活性<sup>[1]</sup>。目前可重构系统的硬件体系结构<sup>[2]</sup>多是采用可重构硬件同一个 GPP 耦合在一起,为使应用的执行时间最短,需要根据应用程序自身的性质,将其合理地划分到软件域和硬件域。

动态可重构系统的软硬件划分问题,相比于传统的软硬件划分问题更为复杂,它需要考虑动态部分的

重构延时,以及硬件任务的并行执行等特征,这实质上是一类 NP 难的组合优化问题<sup>[3]</sup>。目前常见的划分方法主要是启发式搜索算法<sup>[4,5]</sup>,如遗传算法、模拟退火、爬山算法等。虽然这些算法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。

微粒群优化算法<sup>[6-8]</sup> (Particle Swarm Optimization, PSO) 全局搜索能力强,但局部搜索能力往往较差,而混沌优化算法<sup>[9]</sup> (Chaotic Optimization Algorithm, COA) 刚好相反,因此,文中考虑将 PSO 与 COA 相结合,采用 PSO 进行全局搜索,采用 COA 进行局部搜索,两者互为补充,有利于对优良解的局部改良,提高了算法跳出局部最优的能力,以期更好地解决动态可重构系统软硬件划分问题。

## 1 可重构系统软硬件划分问题描述

### 1.1 目标平台

目标平台参考文献[10]中的动态重构模型如图1

收稿日期:2011-11-06;修回日期:2012-02-07

基金项目:国家自然科学基金(61072047)

作者简介:张亮忠(1985-),男,硕士生,主要研究方向为嵌入式系统分析与设计;熊选东,研究员,硕士生导师,主要研究方向为系统工程。

所示,微处理器、可重构硬件、存储器通过系统总线耦合在一起,为了将 GPP 从可重构硬件的配置任务中解脱出来,增加一个配置协处理器负责对可重构硬件进行配置,对于划分到可重构硬件上执行的任务,构建其计算电路所需要的配置 context 保存在一个单独的配置存储器中。为便于微处理器核与可重构硬件并行执行任务,系统中的存储器提供两个独立的访存端口,每个端口都可独立进行读写操作。另外,文中采用一维变宽的动态可重构硬件布局模式,即当任务需要布局到可重构硬件上时,沿水平方向进行1维变宽布局,每个任务都占据可重构硬件的整个高度,在水平方向上根据任务的大小分配合适的宽度,并且任务布局可以和其他任务的执行同时进行,即动态(部分)可重构。

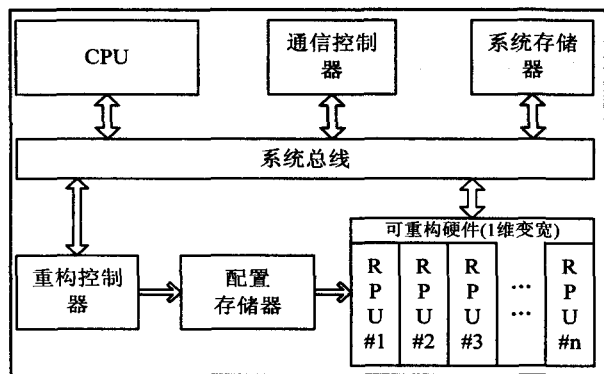


图1 目标平台

## 1.2 任务模型

在可重构系统中,应用程序可以被刻画成离散事件类模型或任务前驱图模型。文中采用任务前驱图即有向无环图(Directed Acyclic Graph, DAG)描述应用程序中任务间的数据/控制依赖关系以及每个任务的成本信息,包括任务的软硬件开销及重构开销等。DAG定义为一个二元组 $\langle V, E \rangle$ ,节点表示任务,包含其软件、硬件成本信息,边表示任务间的依赖关系,图2是用DAG描述的一个拥有10个节点的任务前驱图。

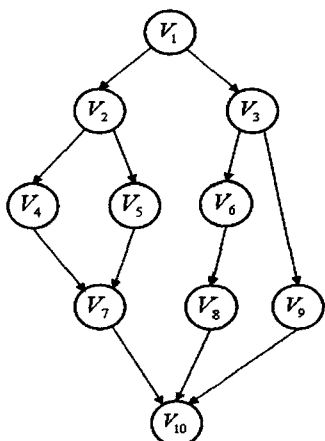


图2 任务前驱图示例

其中  $V = \{V_i | i = 1, \dots, N\}$  表示任务集合,  $|V| = N$

为图中节点(任务)的个数,  $E = \{(V_i, V_j) | 1 \leq i < j \leq N\}$  表示有向边集合,  $|E| = M$  为图中边的个数,边  $(V_i, V_j)$  表示任务之间的数据依赖关系,即  $V_i$  执行完毕是执行  $V_j$  的必要条件。一个具体的任务  $V_i$  可以描述为一个五元组  $\langle t_i^{SW}, t_i^{HW}, t_i^{config}, w_i, h_i \rangle$ ,  $t_i^{SW}$  表示任务  $V_i$  采用软件实现时的执行时间,  $t_i^{HW}$  表示任务  $V_i$  采用硬件实现时的执行时间,  $t_i^{config}$  表示任务  $V_i$  布局到可重构硬件上所花费的时间即重构时间,  $w_i, h_i$  分别表示任务  $V_i$  在可重构硬件上占据的宽度和高度。相对于任务的计算时间,任务间通信时间较小,为简化模型,文中不考虑通信时间,因此假设任务间的通信时间为0。

## 1.3 划分目标

可重构系统的软硬件划分就是去发现由任务模型描述的应用到目标平台的时空映射,通过选择哪些节点使用硬件执行,哪些节点使用软件执行,确定硬件任务的配置顺序和软件任务的执行顺序,以使整个应用的执行时间最小化。

## 2 算法的基本思想

### 2.1 PSO 算法原理

基于其独特的搜索机制,PSO 首先在可行解空间和速度空间随机初始化一个微粒群,群中的每个微粒的位置都代表极值优化问题的一个潜在可能的最优解,通过用坐标位置表征问题解,再通过以位置为参数的目标函数计算得到各微粒的适应度值,其值的好坏表示微粒位置的优劣。每个微粒在解空间中根据速度公式运动,运动过程中通过记录个体最优位置  $P_{best}$  和群体最优位置  $G_{best}$  更新个体位置,  $P_{best}$  代表每个微粒所经历的位置中所对应的适应度值最好的位置,  $G_{best}$  代表所有微粒所经历的所有位置中适应度值最好的位置。微粒的位置每更新一次,就重新计算一次适应度值,然后将新的适应度值与  $P_{best}$ 、 $G_{best}$  的适应度值进行比较,不断更新  $P_{best}$  和  $G_{best}$ 。基本定义:

令  $X = (X_1, X_2, \dots, X_n)$  为  $D$  维空间一个微粒群,一个  $D$  维向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$  表示第  $i$  个微粒在  $D$  维搜索空间中的位置,  $x_i$  表征了问题的一个潜在解。以  $x_i$  为参数,通过目标函数计算微粒  $X_i$  在位置  $x_i$  处的适应度值。  $v_i = (v_{i1}, v_{i2}, \dots, v_{id})^T$  表示微粒  $X_i$  的速度,  $p_i = (p_{i1}, p_{i2}, \dots, p_{id})^T$  为  $X_i$  的个体极值(适应度值最好的位置),  $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})^T$  为  $X$  的全局极值(适应度值最好的  $p_i$ )。

在各次迭代中,微粒通过公式(1)和(2)更新自身的速度和位置:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k), v \in [-v_{max}, v_{max}] \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, x \in [-x_{\max}, x_{\max}] \quad (2)$$

其中,  $i=1, 2, \dots, n; d=1, 2, \dots, D; v_{id}$  表示微粒的速度;  $k$  代表当前迭代次数;  $r_1, r_2$  为介于  $[0, 1]$  之间的随机数。  $c_1, c_2$  都是非负常数, 称为加速度因子。

$\omega$  为惯性权重因子,  $\omega$  越大, 越有利于跳出局部最优, 便于全局搜索;  $\omega$  越小, 越有利于进行精确局部搜索, 便于算法收敛。因此, 文中采用了 (3) 式所示的线性递减惯性权重<sup>[11]</sup>, 随着迭代的进行  $\omega$  逐步变小。

$$\omega(k) = \omega_{\text{start}} - \frac{\omega_{\text{start}} - \omega_{\text{end}}}{k_{\max}} \times k \quad (3)$$

式中,  $k$  为当前迭代次数;  $k_{\max}$  为最大迭代次数,  $\omega_{\text{start}}$  为惯性权重因子的初始值;  $\omega_{\text{end}}$  为  $k$  达到  $k_{\max}$  时的惯性权重因子。

## 2.2 基于实数的编码

本节针对可重构系统软硬件划分问题, 给出了一种基于实数的编码方法。由于标准 PSO 算法中微粒位置为连续值, 因此构造从微粒连续位置到任务执行方式的合适映射是应用标准 PSO 算法解决软硬件划分问题的关键。另外, 鉴于可重构系统软硬件划分过程中任务的组合特性, 需要考虑编码的 Lamarkian 特性<sup>[12]</sup>, 即在所设计的编码方式下, 个体的优良信息是否可通过微粒群进化操作传递给后代个体。为保证这一特性, 文中采用以下的基于实数的编码方式。

由于每个微粒都代表划分问题的一个潜在最优解, 每个微粒的维数即是任务模型 DAG 中节点的个数, 微粒每一个位置坐标  $x_{id}$  对应 DAG 中的一个任务节点,  $x_{id}$  的取值代表节点的实现方式,  $x_{id} < 0$  表示该节点由微处理器核 GPP 采用软件方式实现,  $x_{id} \geq 0$  表示该节点由可重构硬件采用硬件方式实现。

譬如, 考虑图 2 中所示的 10 个节点的划分问题, 假设第  $k$  个微粒的位置为:

$$x_k = (2.2, 3.5, -1.2, 0.4, 4.1, 2.9, -3.2, 3.8, -2.7, 0)^T$$

则对应任务  $V_1 \sim V_{10}$  的执行方式为硬件, 硬件, 软件, 硬件, 硬件, 硬件, 软件, 硬件, 软件, 硬件。显然, 上述编码过程简便, 无需修改标准 PSO 操作就可以对微粒位置实施微粒群优化, 而且能够保证新解的合法性以及解空间的完备性。

## 2.3 适应度值的计算

每个微粒的位置代表一种划分方案, 微粒的适应度即为在该划分方案下任务图 DAG 的完成时间, 完成时间越短, 则微粒的位置越好。在计算适应度时, 文中采用了静态优先级表调度算法, 通过对任务前驱图 DAG 的一次逆向遍历, 按照下式确定每个节点的调度优先级  $P(V)$ :

$$P(V) =$$

$$\begin{cases} 0, V \text{ 在 DAG 中没有后继节点} \\ \text{Max}\{P(V')\} + 1, V' \text{ 是 } V \text{ 的直接后继节点} \end{cases} \quad (4)$$

虽然分配到硬件域的任务可能并行执行, 但它们在配置时是按照 (4) 式确定的优先级在可重构硬件上顺序进行配置, 据此可以计算每个硬件任务的配置完成时刻  $Ct(V)$ :

$$Ct(V_j) = \begin{cases} V_i \text{ 是比 } V_j \text{ 早一步配置的节点,} \\ Ct(V_i) + t_j^{\text{config}} \text{ 并且可重构硬件上有足够的} \\ \text{空间容纳 } V_j \\ \\ V_k \text{ 是先于 } V_j \text{ 布局的节点,} \\ Ft(V_k) \text{ 是节点 } V_k \text{ 的计算} \\ Ft(V_k) + t_j^{\text{config}} \text{ 完成时刻, 并且当 } V_k \text{ 计算完成} \\ \text{释放资源后刚好可容纳 } V_j \end{cases} \quad (5)$$

现在通过对任务图 DAG 的一次正向遍历, 可以计算每个节点的计算完成时刻  $Ft(V)$ , 当节点  $V_i$  被分配到软件域执行时, 其完成时刻等于其所有前驱节点完成时刻同在节点  $V_i$  之前在软件域执行的节点  $V_j$  的完成时刻进行比较, 取最大值再加上  $V_i$  的计算时间  $t_i^{\text{sw}}$ ; 当节点  $V_i$  被分配到硬件域执行时, 其完成时刻等于其所有前驱节点完成时刻同节点  $V_i$  的配置完成时刻  $Ct(V_i)$  进行比较, 取最大值再加上  $V_i$  的计算时间  $t_i^{\text{hw}}$ 。计算公式如下:

$$\begin{aligned} & \text{if } V_i \in V^{\text{sw}} \\ & \quad Ft(V_i) = \text{Max}\{Ft(V_j), \text{Max}(Ft(V_k))\}; \\ & \quad // \text{其中 } V_j \text{ 为在 } V_i \text{ 之前在软件域执行的节点,} \\ & \quad // V_k \text{ 为 } V_i \text{ 的直接前驱节点} \\ & \text{if } V_i \in V^{\text{hw}} \\ & \quad Ft(V_i) = \text{Max}\{Ct(V_i), \text{Max}(Ft(V_k))\}; \\ & \quad // \text{其中 } V_k \text{ 为 } V_i \text{ 的直接前驱节点} \end{aligned} \quad (6)$$

因此, 整个应用的完成时间等于 DAG 中最后一个节点的完成时刻  $Ft(V_N)$ 。

## 3 混沌 PSO

以上给出了应用 PSO 算法解决划分问题的基本思想, 但标准 PSO 算法还存在容易陷入局部极小解、参数依赖性强、鲁棒性差等缺陷。而 COA<sup>[8]</sup> 具有随机性、遍历性、规律性的特点, 可避免搜索过程陷入局部最优, 得到最优解。因此, 本节考虑将两种算法进行融合, 优势互补, 提出一种混沌 PSO (PSOCOA) 算法, 其中 PSO 算法主要用于全局搜索, 而 COA 则根据 PSO 算法的结果执行局部搜索, 以获得可重构系统软硬件

划分问题的近似最优解或满意解。

PSOCOA 划分算法的步骤如下:

步骤 1 初始化:

步骤 1.1 根据任务模型 DAG 的规模,确定初始种群中微粒的个数  $n$  和每个微粒的维数  $D$ ;

步骤 1.2 在满足上下限约束的区间内对每个微粒随机初始化位置矢量  $x_i$  和速度矢量  $v_i$ ;

步骤 1.3 根据位置矢量  $x_i$  确定 DAG 中每个节点的执行方式,按照式(4)计算每个节点的优先级,再根据优先级遍历 DAG 图,按照式(5)和(6)计算得到任务图的完成时刻  $Ft(V_N)$ ,以此作为微粒的适应度值,比较各个微粒的适应度值,选择具有最佳适应度值的微粒位置作为初始  $p_g$ ;

步骤 1.4 将每个微粒自身位置  $x_i$  作为其初始  $p_i$ 。

步骤 2 采用 PSO 进行全局搜索:

步骤 2.1 比较群体中各微粒的适应度值,更新最佳群体位置  $p_g$ ;

步骤 2.2 对于每个微粒,通过比较其经历过的位置,更新各微粒自身的最佳位置  $p_i$ ;

步骤 2.3 对于每个微粒,根据式(1)~式(3)来更新其速度  $v_i$  和位置  $x_i$ ;

步骤 2.4 对于每个微粒,计算其新位置对应的适应度值。

步骤 3 对于群体中的最佳微粒,采用 COA 进行局部搜索:

步骤 3.1 令  $h=0$ ,将决策变量  $x_d^h, d=1,2,\dots,D$  按式(7)映射为 0 到 1 之间的混沌变量  $y_d^h$ ;

$$y_d^h = \frac{x_d^h + x_{\max}}{2x_{\max}}, d=1,2,\dots,D \quad (7)$$

其中,  $h$  为混沌搜索的当前迭代次数,  $x_d^h$  对应 PSO 算法中微粒的第  $d$  维坐标,  $x_{\max}$  对应(2)式中的  $x_{\max}$ 。

步骤 3.2 根据式(8)进行迭代计算,得到  $y_d^{h+1}$ ;

$$y_d^{h+1} = 4y_d^h(1 - y_d^h), d=1,2,\dots,D \quad (8)$$

步骤 3.3 根据上一步计算得到的  $y_d^{h+1}$  按式(9)计算得到  $x_d^{h+1}$ ;

$$x_d^{h+1} = x_{\max} + 2x_{\max}y_d^{h+1}, d=1,2,\dots,D \quad (9)$$

步骤 3.4 根据上一步计算得到的  $x_d^{h+1}, d=1,2,\dots,D$ , 计算其适应度值;

步骤 3.5 若新得到的适应度值优于初始解,则将新解作为 COA 的搜索结果,否则令  $h=h+1$  并返回步骤 3.2;

步骤 3.6 若混沌搜索步数已达到预先设定的最大值,则退出 COA 搜索,保持初始解不变。

步骤 4 若群体最佳位置  $p_g$  的适应度值连续  $L$  步不

变或 PSO 算法的迭代次数  $k > k_{\max}$  则终止算法,输出  $p_g$  对应的划分方案及适应度值,否则返回步骤 2。

## 4 仿真分析

假设目标平台中可重构硬件的宽度为 100, 采用 MATLAB7.01 为仿真平台, 实验环境为 Intel Core 2.2 GHz 处理器, 2Gbyte 内存。使用 TGFF 工具随机生成 5 种不同节点数的 DAG 任务模型, 对 DAG 中每个节点随机生成其软硬件执行时间、重构时间及在可重构硬件上所占的宽度, 各节点的硬件执行时间为软件执行时间的 20%~50%, 重构时间为软件执行时间的 30%~50%, 每个节点在可重构硬件上执行时所占的宽度为 10~50。将文中的 PSOCOA 算法与标准 PSO 算法和遗传算法 (Genetic Algorithm, GA) 的划分结果进行比较, 在 PSO 和 PSOCOA 算法中, 设定种群规模为 30,  $\omega_{\text{start}}=0.9, \omega_{\text{end}}=0.4, c_1=c_2=2, x_{\max}=5, v_{\max}=1$ 。GA 的种群数为 100, 交叉和变异概率分别设为 0.8 和  $1/D$ , 其中  $D$  为 DAG 中节点个数。

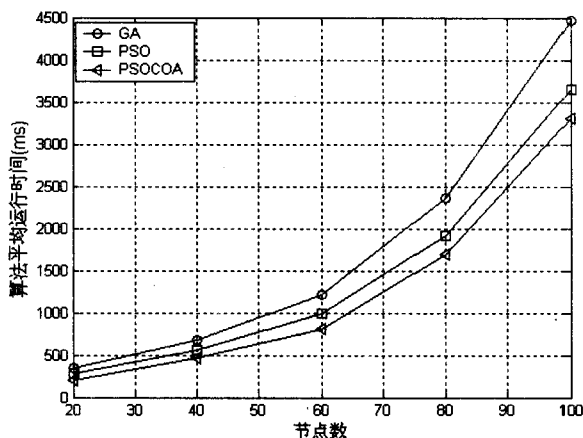


图3 算法平均运行时间对比

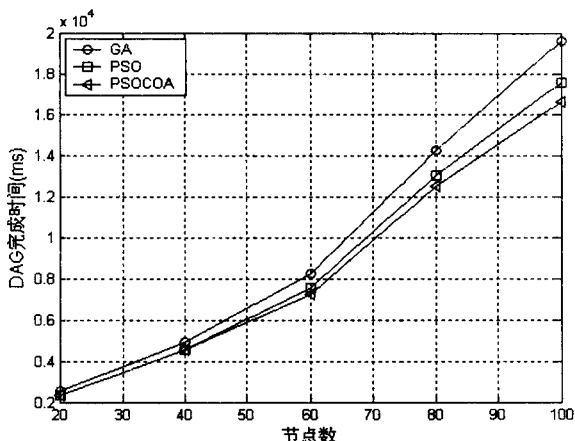


图4 任务图完成时间对比

图3, 图4 为三种算法在 5 种不同节点数情况下, 各进行 30 次实验, 取平均值的实验结果对比。图3 为

(下转第 52 页)

## 4 结束语

文中提出了基于 EPICENTRE 数据模型的海洋钻探数据库的构建方法,实现了海洋钻探业务与 EPICENTRE 数据模型的关联,该方法得到的海洋钻探数据库是面向对象的,与传统的关系数据库相比具有良好的可扩展性、稳定性以及关联性高等优点。

EPICENTRE 数据模型只是逻辑模型,并不能存储数据,可以利用模型投影工具投影成物理模型<sup>[12]</sup>,将关联得到的业务单元模型进行 SQL 编译,实现业务数据的存储,同时开发相应的管理程序和应用程序就可以实现数据的管理与应用。

实现基于 EPICENTRE 数据模型的海洋钻探数据库构建对整个海洋地质调查数据库的建设具有指导意义,未来可以对 EPICENTRE 数据模型进行合理的扩展和修订,使之成为符合海洋地质调查需求的数据模型,为海洋地质调查提供高效实用的数据服务。

### 参考文献:

- [1] 高倩,王新栋,许皓月,等.面向对象的数据库应用研究[J].河北农业科学,2009,13(8):153-156.
- [2] 左凤朝,王文德.面向对象数据模型的研究[J].计算机工程与应用,2001(16):110-112.
- [3] 胡长军,全兆岐,宫发明. EPICENTRE 数据模型应用研究

[J].石油大学学报(自然科学版),2000,24(5):87-91.

- [4] Chalon,Philippe,Karcher,et al. An Innovative Data Modeling Methodology Applied to Petroleum Engineering Problems [C]//Society of Petroleum Engineers. New Orleans:Petroleum Computer Conference,1993:135-143.
- [5] Kim W. Object-orient Database Systems: Promises, Reality and Future [C]//Proceedings of 19th VLDB. [s.l.]:[s.n.],1993:676-687.
- [6] 袁满,张连滨,郭立君,等. EPICENTRE 的核心模型及其数据映射技术[J].大庆石油学院学报,2000,24(4):52-56.
- [7] 袁满. POSC 数据模型与我国石油工业标准化[EB/OL]. 2009. <http://wenku.baidu.com/view/55014b27a5e9856a5612603b.html>.
- [8] 戴勤奋,魏合龙,苏国辉,等.区域海洋地质调查数据库结构模型[J].计算机应用研究,2004,21(3):65-67.
- [9] POSC. Energistics E & P business process reference model [R]. [s.l.]:POSC,2009.
- [10] 王海平,葛珺,王娟.基于 POSC 的油田业务分析与建模方法[J].计算机系统应用,2010,19(3):100-102.
- [11] 袁满,文必龙,张林,等. EPICENTRE 数据模型应用标准及其实现[J].大庆石油学院学报,1998,22(3):40-42.
- [12] 赵海峰,刘一娜,杨国强.基于 EPICENTRE 的数据模型投影工具研究与实现[J].计算机与现代化,2009(8):90-92.

(上接第 48 页)

划分算法的平均运行时间对比反映了划分效率高,图 4 为划分后任务图完成时间对比反映了划分效果优劣。从图中可以看出文中算法在解决动态可重构系统软硬件划分时比标准 PSO 算法和 GA 具有更好的有效性和快速性。

## 5 结束语

文中针对动态可重构系统软硬件划分,提出了一种基于实数编码的混沌微粒群算法。它结合了微粒群优化和混沌优化的优点,弥补了二者的缺陷,具有较强的全局搜索和跳出局部最优的能力。通过与标准 PSO 算法和 GA 算法进行比较,表明文中算法更接近于最优解,求解时间较短,能够更好地解决软硬件划分问题。下一步将继续对适应度计算部分进行改进,以期获得更好的效果。

### 参考文献:

- [1] Hartenstein R. A decade of reconfigurable computing: a visionary retrospective [C]//Proceedings of Design, Automation and Test in Europe. [s.l.]:[s.n.],2001:642-649.
- [2] 姜晶菲,唐玉华,刘福东,等.可重构异构系统结构研究[J].计算机技术与发展,2009,19(1):25-28.

- [3] Gareym R,Johnson D S. Computers and intractability: a guide to the theory of NP-completeness [M]. [s.l.]:W. H. Freeman & Co,1990.
- [4] 李涛,杨愚鲁,马平,等.基于遗传算法的可重构系统软硬件划分[J].计算机工程与应用,2007,43(26):56-58.
- [5] 姚放吾,卢昭材.基于权重可变免疫算法的动态可重构任务划分[J].计算机技术与发展,2009,19(7):52-55.
- [6] 苏守宝,汪继文,方杰.粒子群优化技术的研究与应用进展[J].计算机技术与发展,2007,17(5):249-253.
- [7] 唐俊. PSO 算法原理及应用 [J]. 计算机技术与发展,2010,20(2):213-216.
- [8] 王凌,刘波.微粒群优化与调度算法 [M]. 北京:清华大学出版社,2008.
- [9] 王凌,郑大钟,李清生.混沌优化方法的研究进展[J].计算机技术与自动化,2001,20(1):1-5.
- [10] 沈英哲.可重构计算系统中软硬件代码划分技术研究 [D]. 合肥:中国科学技术大学,2007.
- [11] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//Proc. of the IEEE CEC. [s.l.]:[s.n.],1998:69-73.
- [12] Whitley D, Gordan V, Mathias K. Lamarckian evolution, the baldwin effect and function optimization [C]//Parallel solving from nature-PPSN III. Berlin:Springer,1994:6-15.