基于0-1规划的软硬件划分方法研究

江开忠[1,2]，吕 钊[1]，孙树峰[3]

(1. 华东师范大学计算机系 上海 普陀区 200062; 2. 上海工程技术大学基础学院 上海 松江区 201620;
3. 上海公安高等专科学校学报编辑部 上海 长宁区 200336)

【摘要】将0-1规划算法应用于软硬件协同划分过程中，一种节点的映射方式对应单位立方体上的一个顶点。利用单位立方体几何性质求出无约束的目标最优解；由此解出发，利用折半查找和一定的编码策略向外搜索，直到满足系统约束为止；利用仿真数据对该算法进行了有效性验证。仿真结果表明，0-1算法的收敛速度与遗传算法相当；精度与整线性规划相当。

关 键 词 0-1规划; 软硬件划分; 软硬协同设计
中图分类号 TP393 文献标识码 A

Research on the Application of 0-1 Algorithm in HW/SW Partition

JIANG Kai-zhong[1, 2]，LÜ Zhao[1]，SUN Shu-feng[3]

(1. Department of Computer Science and Technology, East China Normal University Putuo Shanghai 200062;
2. College of Fundamental Studies, Shanghai University of Engineering Science Shongjiang Shanghai 201620;
3. Editorial Board of Journal of Shanghai Public Security Academy Changning Shanghai 200336)

Abstract This paper proposes an innovative 0-1 partitioning algorithm (named 0-1 algorithm) over IP cores which can efficiently partition an expected system into hardware or software parts. The correctness of the 0-1 algorithm is verified. It is illustrated that the result of optimization using the algorithm is better than using genetic algorithm, with similar convergence speed. In addition, this algorithm has the advantage that its convergence is quicker than the Integer Linear Programming (ILP) approach.

Key words 0-1 algorithm; hardware-software co-partition; hardware-software co-design

软硬件协同划分是指在设计系统时确定各个模块是采取软件还是硬件的实现方式，应解决节点的映射方式问题，使系统在满足约束的条件下性能最优。即所占用的软硬件面积最小或功耗最小或运行的时间最小，这是一个NP难题。软硬件协同划分是软硬件协同设计的关键技术，需要解决的问题有：如何兼顾系统的速度和成本，达到成本和性能的最佳结合[1]；确定划分所基于的粒度大小。粒度太粗，算法简单，但结果不好；粒度太细，探索解空间太大，很难找到优化解。

许多研究机构在嵌入式系统协同设计领域做了大量工作，提出了很多软硬件协同划分方法，在划分粒度、划分过程的自动化程度、评价函数以及划分算法选择等方面各有特点[2]。对于搜索最优解的算法有整数规划、混合线性规划、启发式算法以及算法之间的融合等[3-6]。本文提出基于整数规划(ILP)[7]将0-1规划算法应用于软硬件协同划分过程中。该算法取得了较好的效果，可以在合理的时间内为软硬件划分找到最满意的解决方案。

1 SoC系统模型

本文选用固IP生成一个符合互联标准的IP库，每一个IP包括分别由软硬件实现时的软面积和硬面积、软时间和硬时间、软功耗和硬功耗、最大工作频率等参数，是IP核复用技术的基础。一个系统先经过系统工程师进行需求分析，将其划分为适当粒度大小的各功能块，然后依据IP库选用合适的IP核来完成功能。描述系统的任务流图如图1所示，任务流图是有向无环图，只有一个起始节点和一个终止节点，两个节点之间最多只有一条边。节点表示系统的一个任务，边表示任务之间的控制关系或数据流向，每条边的终点任务必须在此边的始点任务完成后才可以开始执行。每个节点包含其软件、硬件代价信息。系统的软硬件划分成为任务流图中的IP核节点向软硬件的映射，即是IP核的软硬件划分。

收稿日期：2005 – 06 – 13
作者简介：江开忠(1965 –)，男，博士，主要从事软硬件协同设计、知识发现、流媒体技术等方面的研究.

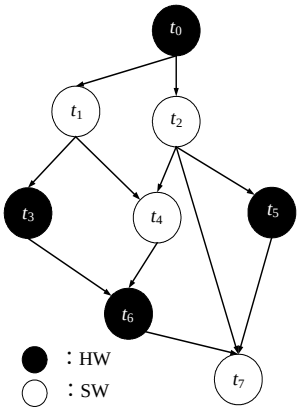

图1　任务流图

2　基于0-1规划的软硬件划分

2.1　系统数学模型

根据任务流图，本文选用的IP核以及对系统在时间面积功耗等方面的总约束，可以抽象出本系统的数学模型。设系统共有N个节点，对节点i从IP库中提取性能特征参数；设节点i的硬时间为t_{hi}，软时间为t_{si}，硬面积为a_{hi}，软面积为a_{si}，硬功耗为w_{hi}，软功耗为w_{si}。

定义 1　映射规则　$x_i=0$ 表示节点 i 映射为硬件；$x_i=1$ 表示 i 映射为软件；$X=(x_1,x_2,\cdots,x_N)$ 表示系统的一个划分。

定义 2　$\mathrm{index0}(X)=\{i\mid x_i=0,i=1,2,\cdots,N\}$ 表示划分 $X=(x_1,x_2,\cdots,x_N)$ 的硬件节点索引集；$\mathrm{index1}(X)=\{i\mid x_i=1,i=1,2,\cdots,N\}$ 表示划分 $X=(x_1,x_2,\cdots,x_N)$ 的软件节点索引集。

设$t_i=t_{si}-t_{\mathrm{hi}}$、$a_i=a_{si}-a_{\mathrm{hi}}$、$w_i=w_{si}-w_{\mathrm{hi}}$、$t_{\max}$、$a_{\max}$和$w_{\max}$分别为系统对时间、面积和功耗的约束；$t_{\max}-\sum_{i=1}^{N}t_{\mathrm{hi}}=t_{\mathrm{win}}$、$a_{\max}-\sum_{i=1}^{N}a_{\mathrm{hi}}=a_{\mathrm{win}}$、$w_{\max}-\sum_{i=1}^{N}w_{\mathrm{hi}}=w_{\mathrm{win}}$，则系统优化模型为：

$$\min\ z(X)\quad \text{s.t.}\ \sum_{i=1}^{N}t_ix_i\le t_{\mathrm{win}}$$

$$\sum_{i=1}^{N}a_ix_i\le a_{\mathrm{win}},\ \sum_{i=1}^{N}w_ix_i\le w_{\mathrm{win}}$$

$$x_i=0\ \text{或}\ 1$$

式中　$z(X)=\sum_{i=1}^{N}(\lambda_{\mathrm{T}}+\lambda_{\mathrm{A}}+\lambda_{\mathrm{W}})x_i=\sum_{i=1}^{N}(s_ix_i)$；$s_i=\lambda_{\mathrm{T}}t_i+\lambda_{\mathrm{A}}a_i+\lambda_{\mathrm{W}}w_i$；$\lambda_{\mathrm{T}}$、$\lambda_{\mathrm{A}}$、$\lambda_{\mathrm{W}}$为用户对时间面积和功耗的正权系数，满足$\lambda_{\mathrm{T}}+\lambda_{\mathrm{A}}+\lambda_{\mathrm{W}}=1$。

2.2　最优解的搜索

2.2.1　预备知识

(1) N维超平面P: $\boldsymbol{S}\cdot\boldsymbol{X}=z$，其中，$\boldsymbol{S}$为$P$的法向量；$z$为常数。$\boldsymbol{T}=\{t_1,t_2,\cdots,t_N\}$，$\boldsymbol{A}=\{a_1,a_2,\cdots,a_N\}$，$\boldsymbol{W}=\{w_1,w_2,\cdots,w_N\}$，则$P_{\mathrm{T}}$: $\boldsymbol{T}\cdot\boldsymbol{X}=t_{\mathrm{win}}$、$P_{\mathrm{A}}$: $\boldsymbol{A}\cdot\boldsymbol{X}=a_{\mathrm{win}}$、$P_{\mathrm{W}}$: $\boldsymbol{W}\cdot\boldsymbol{X}=w_{\mathrm{win}}$分别为时间面积和功耗约束平面。

(2) N维直线，$l:\boldsymbol{X}=\boldsymbol{X}_0+\boldsymbol{S}^0t\quad t\in R$。

(3) N维球面：

$$\Sigma:\left(y_1-\frac{1}{2}\right)^2+\left(y_2-\frac{1}{2}\right)^2+\cdots+\left(y_N-\frac{1}{2}\right)^2=\frac{N}{4}$$

球心$O_\Sigma:\left(\frac{1}{2},\frac{1}{2},\cdots,\frac{1}{2}\right)$，其点用$X$表示。

(4) N维单位立方体顶点集，$\Sigma_{0,1}=\{(x_1,x_2,\cdots,x_N)\mid x_i=0\text{或}1,i=1,2,\cdots,N\}$，$\Sigma_{0,1}\subset\Sigma$，其点用 $X_{0,1}$ 表示。

(5) 平面到点的有向距离 $d=\dfrac{\boldsymbol{S}\cdot\boldsymbol{X}-z}{\|\boldsymbol{S}\|}$，平面$P$: $\boldsymbol{S}\cdot\boldsymbol{X}=z$。

2.2.2　最大时间面积功耗取值范围

设：

$$e_{\mathrm{T}}=1/2\sum_{i=1}^{N}(t_{si}+t_{\mathrm{hi}}),\quad e_{\mathrm{A}}=1/2\sum_{i=1}^{N}(a_{si}+a_{\mathrm{hi}})$$

$$e_{\mathrm{W}}=1/2\sum_{i=1}^{N}(w_{si}+w_{\mathrm{hi}}),\ \sigma_{\mathrm{T}}=1/2\sqrt{N(t_1^2+t_2^2+\cdots+t_N^2)},$$

$$\sigma_{\mathrm{A}}=1/2\sqrt{N(a_1^2+a_2^2+\cdots+a_N^2)}$$

$$\sigma_{\mathrm{W}}=1/2\sqrt{N(w_1^2+w_2^2+\cdots+w_N^2)}$$

当且仅当$e_{\mathrm{T}}-\sigma_{\mathrm{T}}\le t_{\max}\le e_{\mathrm{T}}+\sigma_{\mathrm{T}}$，$e_{\mathrm{A}}-\sigma_{\mathrm{A}}\le a_{\max}\le e_{\mathrm{A}}+\sigma_{\mathrm{A}}$，$e_{\mathrm{W}}-\sigma_{\mathrm{W}}\le w_{\max}\le e_{\mathrm{W}}+\sigma_{\mathrm{W}}$时，系统模型才有可行解。

2.2.3　无约束最优解

$\boldsymbol{S}\cdot\boldsymbol{X}=z$ 沿$\boldsymbol{S}^0$方向平移z增大，沿$\boldsymbol{S}^0$相反方向平移z减小。所以要最小化$\boldsymbol{S}\cdot\boldsymbol{X}$，只需尽可能将平面$\boldsymbol{S}\cdot\boldsymbol{X}=z$向$\boldsymbol{S}^0$相反方向平移。当$\boldsymbol{S}\cdot\boldsymbol{X}=z$到达$X_{\Sigma\min}=\left(\frac{-\sqrt{N}s_1^0+1}{2},\cdots,\frac{-\sqrt{N}s_N^0+1}{2}\right)$时，$z$取得最小值；到达$X_{\Sigma\max}=\left(\frac{\sqrt{N}s_1^0+1}{2},\cdots,\frac{\sqrt{N}s_N^0+1}{2}\right)$时取得最大值。

z在$\Sigma_{0,1}$中的最小值点为$X_{\min}=(x_1^0,x_2^0,\cdots,x_N^0)$：当$s_i^0>0$时$x_i^0=0$；当$s_i^0\le0$时$x_i^0=1$，最大值的点为$(1-x_1^0,1-x_2^0,\cdots,1-x_N^0)$。

2.2.4　初始点的确定

设$\boldsymbol{T}=\{t_1,t_2,\cdots,t_N\}$，$\boldsymbol{A}=\{a_1,a_2,\cdots,a_N\}$，$\boldsymbol{W}=\{w_1,w_2,\cdots,w_N\}$；$V_{\mathrm{T}}$: $\boldsymbol{T}\cdot\boldsymbol{X}\le t_{\mathrm{win}}$，$V_{\mathrm{A}}$: $\boldsymbol{A}\cdot\boldsymbol{X}\le a_{\mathrm{win}}$，$V_{\mathrm{W}}$: $\boldsymbol{W}\cdot\boldsymbol{X}\le w_{\mathrm{win}}$；$\Sigma_{\mathrm{T}}=\Sigma\cap V_{\mathrm{T}}$，$\Sigma_{\mathrm{A}}=\Sigma\cap V_{\mathrm{A}}$，$\Sigma_{\mathrm{W}}=\Sigma\cap V_{\mathrm{W}}$；$2\theta_{\mathrm{T}}$、$2\theta_{\mathrm{A}}$、$2\theta_{\mathrm{W}}$分别为以$\Sigma_{\mathrm{T}}$、$\Sigma_{\mathrm{A}}$、$\Sigma_{\mathrm{W}}$为底、$O_\Sigma$为顶点的N维圆锥体的锥顶角；$\theta_{\mathrm{TA}}$、$\theta_{\mathrm{AW}}$、$\theta_{\mathrm{WT}}$分别为向量$-\boldsymbol{T}$与$-\boldsymbol{A}$、

$-\boldsymbol{A}$与$-\boldsymbol{W}$和$-\boldsymbol{W}$与$-\boldsymbol{T}$的夹角；$0< \theta_{\mathrm{T,A,W,TA,AW,WT}} \leq \pi$，$d_{\mathrm{T}}$、$d_{\mathrm{A}}$、$d_{\mathrm{W}}$分别为$P_{\mathrm{T}}$、$P_{\mathrm{A}}$、$P_{\mathrm{W}}$到$O_{\Sigma}$的有向距离；$2\alpha_{\mathrm{T}} = \theta_{\mathrm{TA}} + \theta_{\mathrm{T}} - \theta_{\mathrm{A}}$，$2\alpha_{\mathrm{A}} = \theta_{\mathrm{TA}} + \theta_{\mathrm{A}} - \theta_{\mathrm{T}}$，$k_{\mathrm{T}} = (\cos\theta_{\mathrm{TA}}\cos\alpha_{\mathrm{A}} - \cos\alpha_{\mathrm{T}})\sin^{-2}\theta_{\mathrm{TA}}$，$k_{\mathrm{A}} = (\cos\theta_{\mathrm{TA}}\cos\alpha_{\mathrm{T}} - \cos\alpha_{\mathrm{A}})\sin^{-2}\theta_{\mathrm{TA}}$，$\boldsymbol{X}_{\mathrm{TA}} = \sqrt{N}\ (k_{\mathrm{T}}\boldsymbol{T}^{0} + k_{\mathrm{A}}\boldsymbol{A}^{0})/2$，$d_1 = (d_{\mathrm{A}} - d_{\mathrm{T}}\cos\theta_{\mathrm{TA}})\ \sin^{-1}\theta_{\mathrm{TA}}$，$d_2 = (d_{\mathrm{T}} - d_{\mathrm{A}})\cos\theta_{\mathrm{TA}} \times \sin^{-1}\theta_{\mathrm{TA}}$，$\cos\beta_{\mathrm{T}} = 2d_1(N - 4d_{\mathrm{T}}^2)^{-\frac{1}{2}}$，$\cos\beta_{\mathrm{A}} = 2d_2(N - 4d_{\mathrm{A}}^2)^{-\frac{1}{2}}$，$\gamma_{\mathrm{T}} = \boldsymbol{T}^{0}\boldsymbol{X}_{\mathrm{TA}}^{0}$，$\gamma_{\mathrm{A}} = \boldsymbol{A}^{0}\boldsymbol{X}_{\mathrm{TA}}^{0}$，$\gamma_{\mathrm{W}} = \boldsymbol{W}^{0}\cdot\boldsymbol{X}_{\mathrm{TA}}^{0}$，$\gamma_{\mathrm{TW}} = \boldsymbol{W}^{0}\cdot\boldsymbol{T}^{0}$，$\gamma_{\mathrm{AW}} = \boldsymbol{W}^{0}\cdot\boldsymbol{A}^{0}$，$\beta_{\mathrm{W}} = \arccos\dfrac{2}{\sqrt{N}}(-d_{\mathrm{T}}\gamma_{\mathrm{T}} + d_1\sqrt{1-\gamma_{\mathrm{T}}^2})$，则有：

(1) $\Sigma_{\mathrm{A}} \subset \Sigma_{\mathrm{T}}$，$\Sigma_{\mathrm{W}} \subset \Sigma_{\mathrm{A}}$，$X_{\mathrm{ch}} = O_{\Sigma} + \sqrt{N}\boldsymbol{W}^{0}/2 \in \Sigma_{\mathrm{T}} \cap \Sigma_{\mathrm{A}} \cap \Sigma_{\mathrm{W}}$。

(2) 其他包含关系类似于(1)。

(3) $\Sigma_{\mathrm{A}} \subset \Sigma_{\mathrm{T}}$，$\Sigma_{\mathrm{W}} \cap \Sigma_{\mathrm{A}} \neq \Phi$，$\Sigma_{\mathrm{W}} \not\subset \Sigma_{\mathrm{A}}$，$\Sigma_{\mathrm{A}} \not\subset \Sigma_{\mathrm{W}} \Rightarrow X_{\mathrm{ch}} = O_{\Sigma} + \sqrt{N}\ (k_{\mathrm{A}}\boldsymbol{A}^{0} + k_{\mathrm{W}}\boldsymbol{W}^{0})/2 \in \Sigma_{\mathrm{T}} \cap \Sigma_{\mathrm{A}} \cap \Sigma_{\mathrm{W}}$，其中，$2\alpha_{\mathrm{A}} = \theta_{\mathrm{AW}} + \theta_{\mathrm{A}} - \theta_{\mathrm{W}}$；$2\alpha_{\mathrm{W}} = \theta_{\mathrm{AW}} + \theta_{\mathrm{W}} - \theta_{\mathrm{A}}$；$k_{\mathrm{A}} = (\cos\theta_{\mathrm{AW}}\cos\alpha_{\mathrm{W}} - \cos\alpha_{\mathrm{A}})\sin^{-2}\theta_{\mathrm{AW}}$；$k_{\mathrm{W}} = (\cos\theta_{\mathrm{AW}}\cos\alpha_{\mathrm{A}} - \cos\alpha_{\mathrm{W}})\sin^{-2}\theta_{\mathrm{AW}}$。

(4) $\Sigma_{\mathrm{T}} \cap \Sigma_{\mathrm{A}} \neq \Phi$，$\Sigma_{\mathrm{T}} \not\subset \Sigma_{\mathrm{A}}$，$\Sigma_{\mathrm{A}} \not\subset \Sigma_{\mathrm{T}}$。该表达式复杂，限于篇幅，本文省略。

2.2.5 搜索系统最优解

定义 3(点运算) $X^{(i)} - X^{(j)} = (x_1^{(i)} - x_1^{(j)},\ x_2^{(i)} - x_2^{(j)}, \cdots, x_N^{(i)} - x_N^{(j)})$，$\left|X_{0,1}\right| = (\left|x_1\right|, \left|x_2\right|, \cdots, \left|x_N\right|)$。

定义 4(点分类) $X^{(i)} = \{(x_1, x_2, \cdots, x_N) \mid x_1 + x_2 + \cdots + x_N = i,\ x_k = 0 \text{ or } 1,\ k = 1,2,\cdots,N\}$，$i = 0,1,\cdots,N$；类$X^{(i)}$中的点用$X_{0,1}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots, x_N^{(i)})$表示。对任意固定的一点$X_{\mathrm{fixed}} \in \Sigma_{0,1}$，$\Sigma_{0,1}$中所有点也可通过$Y_{0,1}^{(i)} = \left|X_{\mathrm{fixed}} - X_{0,1}^{(i)}\right| \in Y^{(i)} \subset \Sigma_{0,1}$分成$N+1$类。

由于在$\Sigma_{0,1}$中使z取得最小值的点不一定满足系统的约束条件，$X_{\min}$不一定为系统的最优解。为了求出系统的最优解，本文以$X_{\min}$为基础、以X_{ch}初始点向外搜索，直至搜索的点满足系统全部约束条件并尽可能靠近$X_{\min}$为止。

设$\Sigma_{\mathrm{T}} \cap \Sigma_{\mathrm{A}} \cap \Sigma_{\mathrm{W}} \neq \Phi$，$C_1(X) = \boldsymbol{T}-\boldsymbol{X}-t_{\mathrm{win}}$，$C_2(X) = \boldsymbol{A}\cdot\boldsymbol{X}-a_{\mathrm{win}}$，$C_3(X) = \boldsymbol{W}\cdot\boldsymbol{X}-w_{\mathrm{win}}$，$X_{\mathrm{opt}}$为所求的最优点，$z_{\mathrm{opt}}$为相应最优值。搜索$X_{\mathrm{opt}}$的算法如下：

$$X_{\mathrm{ch}} = (x_1^{(\mathrm{ch})}, x_2^{(\mathrm{ch})}, \cdots, x_N^{(\mathrm{ch})})$$

For i=1 to N if $x_i^{(\mathrm{ch})} > 1/2$，$x_i^{(\mathrm{ch})} = 1$，else $x_i^{(\mathrm{ch})} = 0$ end if，next i。

$X_{\mathrm{opt}} = X_{\mathrm{ch}}$，$z_{\mathrm{opt}} = z(X_{\mathrm{ch}})$；$X = |X_{\min} - X_{\mathrm{ch}}| = (x_1, x_2, \cdots, x_N)$，$l_2 = \sum_{i=1}^{N} x_i$；$X = X_{\min}$，$l_1 = 0$，$T$=False。

1) If $C_1(X)>0$，goto 2)；else if $C_2(X)>0$，goto 2)；else if $C_3(X)>0$，goto 2)；else $z = z(X)$。If $z< z_{\mathrm{opt}}$，$z_{\mathrm{opt}} = z$，$X_{\mathrm{opt}} = X$，T=True，goto 3)。

2) $l = \left[\dfrac{l_1 + l_2}{2}\right]$；if $l = l_1$ or $l = l_2$，goto (3)；T=False。Let $X_{\mathrm{computed}}^{(l)} = \Phi$，ctrl2=(1)。

(1) if index0($X_{0,1}^{(l)}$) goto (2)，随机选取一点$X_{0,1}^{(l)} \in X^{(l)}$ ($X_{0,1}^{(l)} \notin X_{\mathrm{computed}}^{(l)}$)；ctrl2=(2)，goto (5)。

(2) 变换(1)的$x_i^{(l)} = 1$：if $l \leq N/2$，从index0($X_{0,1}^{(l)}$)中选取l个i置index1($X_{0,1}^{(l)}$)，对所有$i \in$ index1($X_{0,1}^{(l)}$)置$x_i^{(l)} = 0$；否则从index1($X_{0,1}^{(l)}$)中选取$N-l$个i置$x_i^{(l)} = 0$，对所有$i \in$ index0($X_{0,1}^{(l)}$)置$x_i^{(l)} = 1$。ctrl2=(3)，goto (5)。

(3) 变换(1)的$X_{0,1}^{(l)}$：if $l \leq N/2$，从index0($X_{0,1}^{(l)}$)中选取$[l/2]$个i置$x_i^{(l)} = 1$，从index1($X_{0,1}^{(l)}$)中选取$[l/2]$个i置$x_i^{(l)} = 0$；否则从index1($X_{0,1}^{(l)}$)中选取$[(N-l)/2]$个i置$x_i^{(l)} = 0$，从index0($X_{0,1}^{(l)}$)中选取$[(N-l)/2]$个i，并置$x_i^{(l)} = 1$。ctrl2=(4)，goto (5)。

(4) 按(2)的方法变换(3)的点$X_{0,1}^{(l)}$。ctrl2=(1)，goto (5)。

(5) $X_{\mathrm{computed}}^{(l)} = X_{\mathrm{computed}}^{(l)} \bigcup \{X_{0,1}^{(l)}\}$，$X = \left|X_{\min} - X_{0,1}^{(l)}\right|$。

If $C_1(X)>0$，goto ctrl2；else if $C_2(X)>0$，goto ctrl2；else if $C_3(X)>0$，goto ctrl2；else $z = z(X)$。If $z< z_{\mathrm{opt}}$，$z_{\mathrm{opt}} = z$，$X_{\mathrm{opt}} = X$，T=True，goto ctrl2。

(6) If T=True then $l_2 = l$，else $l_1 = l$，goto 2)。

3) 最优解为X_{opt}，对应的最优值为z_{opt}。

2.3 算法分析

对N个节点的系统有2^N种不同软硬件的映射方式，这是一个NP难题。本文通过无约束的最优解搜索有约束的最优解，克服了陷入局部最优的可能性，只需进行加减运算，进一步提高了效率。

在最坏的情况下，进入循环主体的次数为：

$$\sum_{i=1}^{k} C_N^{j_i} \approx O(C_N^{j_1}\log_2 N)$$

式中 $k = [\log_2 N] + 1$；$j_1 = [N/2]$；$j_i = [(2^{i-1}-1)N/2^i]$；$i = 2,3,\cdots,k$。每一次循环至多计算三次条件式$C_i(X)$和一次目标式$z(X)$。若$X_{\min}$是点$((-\sqrt{N}s_1^0 + 1)/2, \cdots, (-\sqrt{N}s_N^0 + 1)/2)$，主体循环中计算体可改为：If $C_1(X)>0$，goto ctrl2；else if $C_2(X)>0$，goto ctrl2；else if $C_3(X)>0$，goto ctrl2；else $z = z(X)$。If $z<z_{\mathrm{opt}}$，$z_{\mathrm{opt}} = z$，

$X_{\text{opt}}=X$，T=True，goto ctrl2。表明一旦 $X^{(l)}$ 中有一点 $X_{0,1}^{(l)}$ 满足所有约束条件，即可跳出本 l 循环。事实上若 $s_1^0(x_1^0-0.5)+\cdots+\ s_N^0(x_N^0-0.5)\to -1$，(5)也可用该计算体近似代替。如果只需一个近似最优解，可将(1)中 $X_{\text{computed}}^{(l)}=X^{(l)}$ 用一记数变量代替，记数变量取100～1 000。

3　实验评价

本文采用800 MHz CPU的PC机、512 MB RAM。因为对每一个l，$X^{(l)}$ 有 C_N^l 个点，当N较大时不可能检查所有的点，所以在实验中 $X_{\text{computed}}^{(l)}=X^{(l)}$ 被一个记数变量取代，并作为参数输入；而IP核的数据由机器随机产生。本文选择了许多参数，包括节点数和记数变量进行计算，如图2所示。从图可知，0-1算法的运行时间几乎与遗传算法(GA)相当，好于整线性规划(ILP)。本文采用的搜索策略除了使用了折半算法，还使用了较好的编码变换，使算法同时具有ILP的高精确性与GA的高效性。运行时间与记数变量的近似于线性的关系，如图3所示。

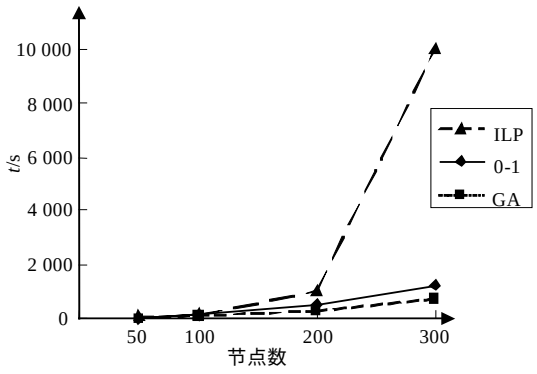

图2　算法比较

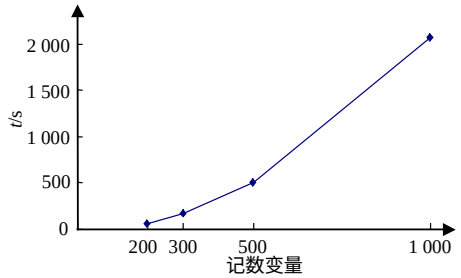

图3　运行时间与记数变量的关系

4　结 束 语

本文研究了基于IP核的SoC协同设计，提出了基于IP核的0-1软硬件划分算法。该算法成功地应用于“iCDMdt Platform”[8]。今后将进一步改进算法，并与其他算法如GA、SA及Tabu进行融合，使算法具有更高的效率和精确性。

本文研究工作得到了上海自然科学基金中德合作软硬协同设计技术项目的支持，感谢顾君忠教授的指导及栾静、黄瑞、程喧的建议。

参 考 文 献

[1] LI Yang-bin, CALLAHAN T, DARNELL E, et al. Hardware/software co-design of embedded reconfigurable arch itectures[C]//Proceedings of Design Automation Conference. Piscataway, USA: IEEE Computer Society, 2000.

[2] EDWARDS S, LAVAGNO L, LEE E A. Design of embedded system: formal mdels, validation and synthesis[J]. Proceedings of the IEEE, 1997, 85 (3): 366-386.

[3] NIEMANN R. Hardware/software co-design for data flow dominated embedded systems[M]. Boston: Kluwer Academic Publishers, 1998.

[4] SAHA D, MITRA R S, BASU A. Hardware software partitioning using genetic algorithm. In: Agrawal V, Mahabala HN, eds[C]//Proc. of the 10th Int. Conf. on VLSI Design. Piscataway: IEEE Computer Society, 1997: 155-160.

[5] PENG Z, KUCHCINSKI K. An algorithm for partitioning of application specific systems. In: Courtois B, eds[C]//Proc. of the European Conf. on Design Automation (EDAC). Piscataway: IEEE Computer Society , 1993: 316-321.

[6] XIONG Zhi-hui, LI Si-kun, CHEN Ji-hua. Hardware/ software partitioning based on dynamic combination of genetic algorithm and ant. algorithm[J]. Journal of Software, 2005, 16(4): 503-512.

[7] ERNST R, HENKEL J, BENNER T. Hardware-software cosynthesis for microcontrollers[J]. IEEE Design & Test of Computers, 1993,10(4): 64-75.

[8] LUAN Jing, GU Jun-zhong. ICDMdt: focused the model mapping and performance optimization in embedded system design[C]//The 5th International Conference on Computer and Information Technology. Piscataway, USA: IEEE Computer Society, 2005.

编　辑　黄　莘