学术论文

集成产品开发规划的任务调度

鲁建厦，陈呈频，兰秀菊，陈　勇，谢列卫
（浙江工业大学工业工程研究所，杭州　310032）

［摘要］　在阐述集成产品开发（IPD）任务特点的基础上，研究了集成产品开发任务的耦合性和层次性，建立了分层分布式规划的策略，并引入代价矩阵概念，给出了任务调度模型和相应的匈牙利算法，并用实例演示了计算过程。

［关键词］　集成产品开发；任务调度；过程规划；目标决策
［中图分类号］ TP39　　［文献标识码］ A　　［文章编号］ 1009－1742（2004）05－0056－05

1　引言

集成产品开发[1,2]（IPD）是一个框架性的概念，这一概念描述了一个优化产品开发过程的集成和并行的、跨学科的、整体的并以人为中心的方法，这种方法基于各种创新技术的应用和计算机及网络平台的方法，其最终目的是提供面向客户需求的数字化产品模型或物理模型，其实质是体现过程、产品、组织和资源的全面集成。集成产品开发充分体现了并行工程的思想，其目标是要求产品开发一次成功（right the first time），这一目标必须通过产品开发过程的优化设计，减少过程的冲突和提高资源的优化配置水平来实现。如何实现产品开发过程的设计，就必须研究集成产品开发过程的规划和优化问题。集成产品开发过程的任务调度是集成产品开发过程规划和优化的关键技术[3]。任务调度的实质是对任务的合理规划和重组，寻找最优的任务排序和对任务进行合理分工，使得产品开发成本最低或时间最短，IPD 任务调度将有助于：**a.** 产品的并行开发；**b.** 任务的合理分配；**c.** 资源的合理利用；**d.** 组织的合理构架。

2　IPD 任务的特点

IPD 的基本组织形态是集成产品小组（IPT），IPT 强调多学科专家的协同工作，充分考虑产品开发中的各种因素，有效地实现知识集成。IPD 工作方式决定了 IPD 任务具有相当的复杂性[2]，这种复杂性主要体现在任务的耦合性和层次性两方面。

IPD 任务的耦合性主要表现为各任务之间的相互依赖和相互制约的关系，这种关系通过任务间的变量来表示，这些变量构成了任务耦合间的信息流。图 1 表示了任务 J_1，J_2，J_3 之间的耦合关系和相互间的约束变量 δ_{12}，δ_{13}，δ_{21}，δ_{23}，δ_{31}，δ_{32}，一般情况下，$\delta_{12}=\delta_{21}$，$\delta_{13}=\delta_{31}$，$\delta_{23}=\delta_{32}$。

IPD 任务的层次性主要表现为 IPD 过程中的 4 个不同层次的任务，它们是不同 IPD 间的任务，同一 IPD 内不同领域间的任务，同一领域中不同分工间的任务以及同一分 工中不同对象间的任务，图 2 表示了某集成产品开发中任务的 4 个不同层次。在以下的讨论中，用任务承担者的概念，来表示不同层次的 IPD 团队、领域专家、设计小组和设计人员。

［收稿日期］　2003－10－29；修回日期　2004－02－08
［基金项目］　国家自然科学基金资助项目（70271032）
［作者简介］　鲁建厦（1963－），男，浙江余姚市人，浙江工业大学副教授

任务	J_1	J_2	J_3
J_1	0	δ_{12}	δ_{13}
J_2	δ_{21}	0	δ_{23}
J_3	δ_{31}	δ_{32}	0

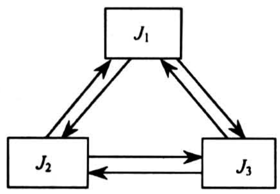

图 1　IPD 任务的耦合性

Fig.1　The coupling of IPD task

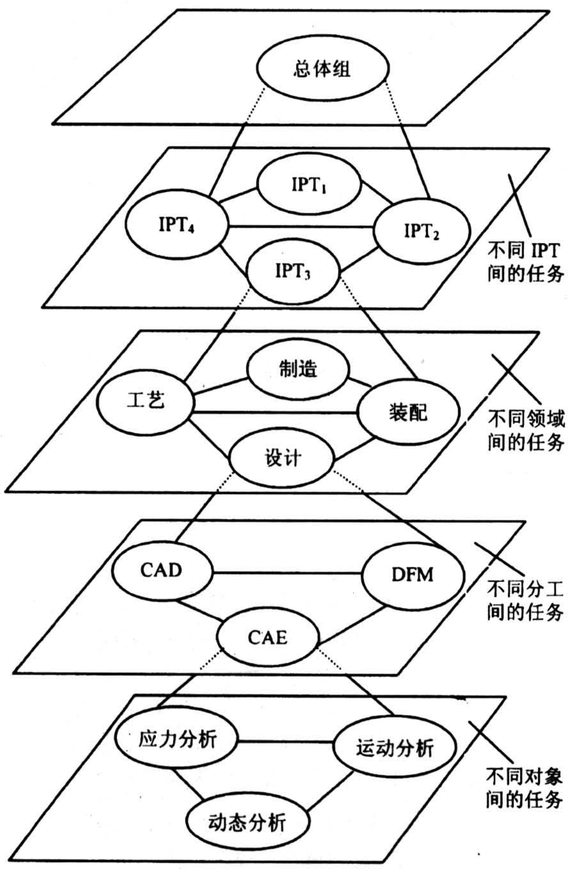

图 2　IPD 任务的层次性

Fig.2　The administrative levels of IPD task

3　分层分布式任务规划策略

IPD 任务的两大特点决定了任务规划的复杂性，在对任务进行优化调度前必须先进行任务的分解和整合。

任务的复杂性与产品的复杂性密切相关，而产品的复杂性取决于产品中各元素间关系的复杂程度。关系描述了如何通过事件之间的相互作用而形成产品的功能，如果在开发过程中明确定义了所有的关系，那么，就可清楚地描述产品结构表中各个事件。图 3 表示了某机器人产品各元素间的关系。

关系在表述对事件的需求时明确了功能联系，如果开发过程顺着关系进行，那么这些联系可持续到将产品开发任务分解到不同的设计人员或供货商。但复杂产品的关系往往存在相互迭代的循环，这些循环只能用附加的知识或经验来解决，一个循环可以覆盖多个任务，从而造成任务规划的难度。

因此，若想使任务规划问题简单化，首先必须使产品开发过程中的各种关系可以把握，使得产品开发尽可能并行进行。根据参考文献［4］提出的设计结构矩阵（DSM，design structure matrices）的思想，提出以下分层分布任务规划的策略。

根据产品的初步设计方案，建立如图 3 所示的关系图形描述，并用矩阵来定义开发和设计任务的顺序，在结构矩阵中标有标记“＊”表示对应的行元素（事件）和列元素（事件）之间的存在着某种关系。

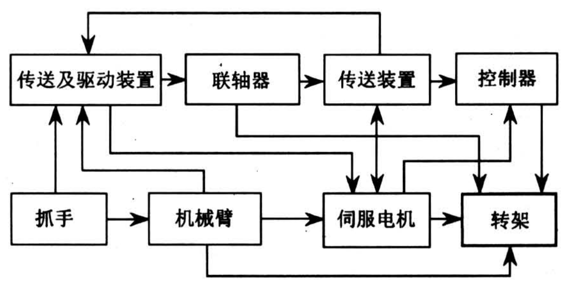

图 3　机器人各部件间的关系

Fig.3　The relationship of robot parts

通过行和列的交换，使结构矩阵可以变换成分块三角矩阵形式，结果主对角线各块之间不再存在循环。由于产品的结构没有变化，所以循环依然存在，只不过这些循环被包含在各块中，而在块之间只存在着偶尔的联系，这种相对集中化是分布式设计的重要前提。对图 3 的机器人设计来说，经变换后，伺服电机、传动装置和传送带 3 个部件被组成 1 个相对独立的块，该块可由某个 IPT 独立承担开发。

在确定了相对独立的 IPT 任务以后，可根据 DMS 的思想，对各个块事件的零件关系进一步地分布式规划，从而确定不同领域间的任务，依次类推，可对不同分工和不同对象间的任务进一步的

规划。

分层分布式任务规划方法可大大简化复杂产品任务规划的复杂性，并使任务执行的并行性大大提高。在此基础上可较容易实现对任务的最佳调度。

4 任务调度模型和算法

4.1 任务调度模型

任务调度的目标是寻求最佳的任务安排，为了不失一般性，在讨论任务调度模型之前，先引入任务代价矩阵的概念。

任务代价矩阵 $\boldsymbol{C} = (\boldsymbol{C}_{ij})$，其中 $\boldsymbol{C}_{ij}$ 表示第 i 个任务承担者 A_i 从事 j 项任务 J_i 的代价，代价可表示 IPD 中的成本、利润、时间和资源利用率等。

当任务规模较小时，其调度可通过对代价矩阵 $\boldsymbol{C}$ 进行转化，将 $\boldsymbol{C}$ 转化成 $\boldsymbol{C}'$，使 $\boldsymbol{C}'$ 的每行每列都有 0 元素，从而通过观察法得到 A_i 和 J_i 之间的最优匹配。除了观察法以外，分支定阶法也是一种对小规模任务调度的有效算法。

当任务规模较大时，必须有相应的算法求解最优匹配。以下给出任务调度的一般模型。

$$\min z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij},$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, i = 1,2,\cdots,n,$$

$$\sum_{j=1}^{n} x_{ij} = 1, j = 1,2,\cdots,n,$$

$$x_{ij} = \begin{cases} 0, \text{第 } j \text{ 项任务不由 } A_i \text{ 做}, \\ 1, \text{其他}。\end{cases}$$

$$i,j = 1,2,\cdots,n。$$

上述任务调度的一般模型，适合于代价最小的任务安排，如成本最低的任务调度、时间最短的任务调度可采用这一模型。但有时需要寻求代价最大的任务安排。如利润最大或资源利用率最高等，此时，上述一般模型的目标将是求最大值，即 $\max z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij}$。

在上述模型中，讨论的是 n 个任务承担者 A_i 完成 n 项任务 J_i 的情况，在实际的 IPD 任务调度中，往往需要调度 m 个任务承担者完成 n 项任务的情况，而且每一任务承担者的代价 $C_{ij}(i = 1, 2, \cdots, m)$ 约束于 $b_i(i = 1, 2, \cdots, m)$，即 $\max\{C_{ij}\} \leqslant b_i$，此时的任务调度模型为：

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij},$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, j = 1,2,\cdots n,$$

$$\sum_{j=1}^{n} C_{ij}x_{ij} \leqslant b_i, i = 1,2,\cdots,m,$$

$$x_{ij} = \begin{cases} 0, \text{第 } j \text{ 项任务不由 } A_i \text{ 做}, \\ 1, \text{其他}。\end{cases}$$

4.2 任务调度算法

对于任务调度的一般模型，给出匈牙利算法。

定义：

1) 二分图 $G_1 = (\boldsymbol{V}, \boldsymbol{E}), \boldsymbol{V} = \{x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\}$，$E = \{(x_i, y_j) |\ b_{ij} = 0, \ i, j = 1, 2, \cdots, n\}$。

2) 二分图 G_1 的顶点 x_i 或 y_j 得到匹配，则称为饱和，否则称为未饱和。

3) 二分图 G_1 中 ν 点的领接点集合，称为 $\boldsymbol{\Gamma}(\nu)$。

4) 若道路 $x_{i1}y_{j1}$，$x_{i1}y_{j1}$，$\cdots$，$x_{ik}y_{jk}$ 的首尾二顶点 x_{i1}，y_{jk} 未饱和，$y_{j1} - x_{i2}$，$y_{j2} - x_{i3}$，$\cdots$，$y_{j(k-1)} - x_{ik}$ 都是匹配边，则称该道路为可增广道路。

5) 满足规则 $1+1=0+0=0$，$1+0=0+1=1$ 的运算称为对称差，记作 $\otimes$。如：

$$\boldsymbol{M} = \{y_{j1}x_{i2}, y_{j2}x_{i3}, \cdots, y_{j(k-1)}x_{ik}\},$$

$$\boldsymbol{E}(p) = \{x_{i1}y_{j1}, y_{j1}x_{i2}, x_{i2}y_{j2}, y_{j2}x_{i3}, \cdots, y_{j(k-1)}x_{ik}, x_{ik}y_{jk}\},$$

$$M \oplus E(p) = \{x_{i1}y_{j1}, y_{j1}x_{i2}, x_{i2}y_{j2}, \cdots, x_{ik}y_{jk}\}。$$

求解任务调度一般模型的算法：

step 1 给初始标号：$l(A_i) \leftarrow \min\{C_{ij}\}$，$i = 1, 2, \cdots, n$；$l(J_i) \leftarrow 0$，$j = 1,2,\cdots,n$。

step2 构造矩阵 $\boldsymbol{B} = (b_{ij})$，$b_{ij} = c_{ij} - l(A_i) - l(J_j)$ $i, j = 1, 2, \cdots, n$；由矩阵 $\boldsymbol{B}$ 的零元素构造二分树 $G_1 = (\boldsymbol{V}, \boldsymbol{E})$。

step 3 为能找到图 G_1 的匹配，从 $\boldsymbol{E}$ 中选出 n 条边使 $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$ 和 $\boldsymbol{Y} = (y_1, y_2, \cdots, y_n)$，每点都与一条边有且只有一条关联，则已达到最优解，到此结束，否则转 step4。

Step 4 找一条饱和点 $x_0 \in \boldsymbol{X}$，作 $\boldsymbol{V}_1 \leftarrow \{x_0\}$，$\boldsymbol{V}_2 \leftarrow \varnothing$。

Step5 若二分图 G_1 有 $\boldsymbol{\Gamma}(\boldsymbol{V}_1) = \boldsymbol{V}_2$，则转 Step6，否则转 Step 7。

Step 6 作 $\alpha \leftarrow \min\{l(x_i) + l(y_i) - C_{ij}\}$，$x_i \in \boldsymbol{V}_1$，$y_i \in \boldsymbol{\Gamma}(\boldsymbol{V}_1) \setminus \boldsymbol{V}_2$，

$$l(v) \leftarrow \begin{cases} l(v) + a, v \in \boldsymbol{V}_1 \\ l(v) - a, v \in \boldsymbol{V}_2 \\ l(v), \text{其他。} \end{cases}$$

构造矩阵 $\boldsymbol{B}_1 = (b_{ij})$ 及二分树 G_1。

Step 7　找 $y \in \boldsymbol{\Gamma}(\boldsymbol{V}1) \setminus \boldsymbol{V}_2$。

Step 8　若 y 饱和则转 Step 9，否则转 Step 10。

Step 9　存在饱和边 xy，$\boldsymbol{V}_1 \leftarrow \boldsymbol{V}_1 \cup \{x\}$，$\boldsymbol{V}_2 \leftarrow \boldsymbol{V}_2 \cup \{y\}$，转 Step 5。

Step10　从 x_0 到 y 有一增广道路 P，作 $\boldsymbol{M} \leftarrow \boldsymbol{M} \oplus \boldsymbol{E}(p)$，转 Step 3。

5　应用举例

已知某产品的利润矩阵，要求在该利润矩阵约束下的最佳任务调度，即寻求利润最大的任务安排方案。利润矩阵

$$\boldsymbol{P} = \begin{bmatrix} 5 & 7 & 7 & 6 & 3 \\ 4 & 4 & 0 & 4 & 4 \\ 4 & 6 & 6 & 3 & 0 \\ 0 & 3 & 3 & 0 & 0 \\ 3 & 4 & 4 & 5 & 5 \end{bmatrix}$$

$\boldsymbol{P}$ 相当于任务决策一般模型中的代价矩阵 $\boldsymbol{C}$，用匈牙利算法求解该决策问题时，将目标的极小值问题转化为极大值问题，其主要步骤如下：

1) 初始标志数

$l(A_i) = \max\limits_j(p_{ij})$，$i = 1, 2, \cdots, n$，$l(J_j) = 0, j = 1, 2, \cdots, n$。保证 $l(A_i) + l(J_j) \geqslant p_{ij}$，构造 $\boldsymbol{B} = (b_{ij})$，其中 $b_{ij} = p_{ij} - l(A_i) - l(J_j)$，初始标志数下的 $\boldsymbol{P}$ 矩阵：

$$\begin{array}{cc} & l(A_i) \\ & \Downarrow \\ \begin{array}{c} A_2 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{array} & \begin{array}{c} 7 \\ 4 \\ 6 \\ 3 \\ 5 \end{array} \begin{bmatrix} 5 & 7 & 7 & 6 & 3 \\ 4 & 4 & 0 & 4 & 4 \\ 4 & 6 & 6 & 3 & 0 \\ 0 & 3 & 3 & 0 & 0 \\ 3 & 4 & 3 & 5 & 5 \end{bmatrix} \\ & \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \Leftarrow l(J_j) \\ J_1 & J_2 & J_3 & J_4 & J_5 \end{array} \end{array}$$

根据初始标志数构造的 $\boldsymbol{B}$ 矩阵：

$$\boldsymbol{B}_1 = \begin{bmatrix} 2 & 0^* & 0 & 1 & 4 \\ 0^* & 0 & 4 & 0 & 0 \\ 2 & 0 & 0^* & 3 & 6 \\ 3 & 0 & 0 & 3 & 3 \\ 2 & 1 & 2 & 0^* & 6 \end{bmatrix}$$

根据 $\boldsymbol{B}_1$ 可得二分树 G_1，求得匹配 $A_1 - J_2$，$A_2 - J_1$，$A_3 - J_3$，$A_5 - J_4$，见 $\boldsymbol{B}_1$ 中带 * 的0元素，二分树的匹配图如图 4 所示，可知 A_4 未匹配，G_1 未饱和。

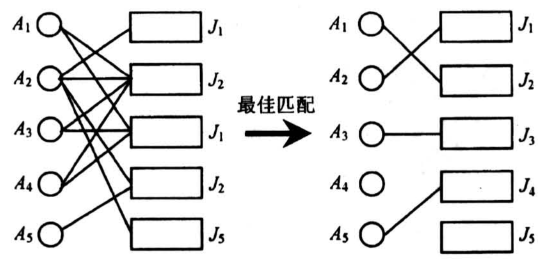

图 4　G_1 的初始匹配

Fig.4　The original matching of G_1

2) 由于 A_4 未匹配，$\boldsymbol{V}_1 \leftarrow \{A_4\}$，$\boldsymbol{V}_2 \leftarrow \varnothing$；$\boldsymbol{\Gamma}(\boldsymbol{V}_1) = \{J_2, J_3\} \neq \boldsymbol{V}_2$，$\boldsymbol{\Gamma}(\boldsymbol{V}_1) \setminus \boldsymbol{V}_2 = \{J_2, J_3\}$，取 J_3 匹配边 $A_3J_3 \in M$，有 $\boldsymbol{V}_1 \in \boldsymbol{V}_1 \cup \{A_3\} = \{A_3, A_4\}$，$\boldsymbol{V}_2 \leftarrow \boldsymbol{V}_2 \cup \{J_3\} = \{J_3\}$，$\boldsymbol{\Gamma}(\boldsymbol{V}_1) = \{J_2, J_3\} \neq \boldsymbol{V}_2$，$\boldsymbol{\Gamma}(\boldsymbol{V}_1) \setminus \boldsymbol{V}_2 = \{J_2\}$；取 J_2 匹配边 $A_1J_2 \in M$，有 $V_1 = \{A_1, A_3, A_4\}$，$V_2 = \{J_2, J_3\}$，$\boldsymbol{\Gamma}(\boldsymbol{V}_1) = \{J_2, J_3\} = \boldsymbol{V}_2$，$\boldsymbol{\Gamma}(\boldsymbol{V}_1) \setminus \boldsymbol{V}_2 = \{J_1, J_4, J_5\}$。

3) $\alpha = \min\{l(A_i) + l(J_i) - p_{ij}\} = 1$，$A_i \in V_1$，$y_j \in \boldsymbol{\Gamma}(\boldsymbol{V}_1) \setminus \boldsymbol{V}_2$。重新更改标志数 $l(A_1) = 6$，$l(A_3) = 5$，$l(A_4) = 2$，$l(J_2) = 1$，$l(J_3) = 1$。在新的更改后标志数下的 $\boldsymbol{P}$ 矩阵：

$$\begin{array}{cc} & l(A_i) \\ & \Downarrow \\ \begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{array} & \begin{array}{c} 6 \\ 4 \\ 5 \\ 2 \\ 5 \end{array} \begin{bmatrix} 5 & 7 & 7 & 6 & 3 \\ 4 & 4 & 0 & 4 & 4 \\ 4 & 6 & 6 & 3 & 0 \\ 0 & 3 & 3 & 0 & 0 \\ 3 & 4 & 3 & 5 & 5 \end{bmatrix} \text{。} \\ & \begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \Leftarrow l(J_j) \\ J_1 & J_2 & J_3 & J_4 & J_5 \end{array} \end{array}$$

根据更改好标志数构造的 $\boldsymbol{B}$ 矩阵：

$$
\boldsymbol{B}_1=\begin{bmatrix} 1 & 0 & 0 & 0^* & 3 \\ 0^* & 1 & 5 & 0 & 0 \\ 1 & 0^* & 0 & 2 & 5 \\ 2 & 0 & 0^* & 2 & 2 \\ 2 & 2 & 3 & 0 & 0^* \end{bmatrix}。
$$

根据 $\boldsymbol{B}_1$ 可得新的二分树 G_1，求得匹配 A_1-J_4，A_2-J_1，A_3-J_2，A_4-J_3，A_5-J_5，见 $\boldsymbol{B}_1$ 中带 * 的0元素，图5表示了新的 G_1 的最佳匹配情况，此时 G_1 饱和。

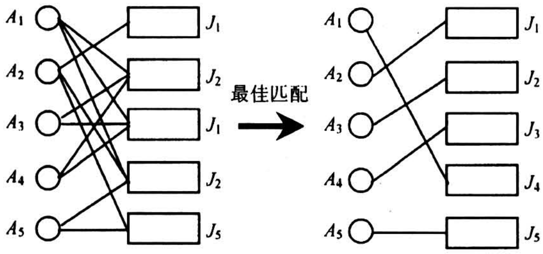

图5 G_1 的最佳匹配

Fig.5 The optimal matching of G_1

4）根据最佳任务0安排，可方便求得该任务安排的利润最大值 $\max z = p_{14}+p_{12}+p_{32}+p_{43}+p_{55}=6+4+6+3+5=24$。

6 结论

集成产品开发过程的任务调度是集成产品开发过程规划和优化的关键技术。IPD的任务具有相当的复杂性，主要体现在任务的耦合性和层次性这两方面。为使任务规划问题简单化，必须使产品开发过程中的各种关系可以把握，产品开发尽可能并行进行。根据设计结构矩阵DMS的思想，提出了分层分布任务规划的策略。建立了分层分布式规划的策略，并引入代价矩阵概念，给出了任务调度模型和相应的匈牙利算法和实例。任务调度的实质是对任务的合理规划和重组，寻找最优的任务排序和对任务进行合理分工，使得产品开发成本最低或时间最短。IPD任务调度研究对IPD过程优化设计、减少IPD的过程冲突和实现资源的优化配置具有重要意义。

参考文献

[1] Vajna S, Burchardt C. Integrated product development [J]. Industrial Engineering and Management, 1999, 4 (3): 40～50

[2] Dik L L. Integrated product and process date management [J]. Integrated Computer-Aided Engineering, 1996, 3 (1): 1～4

[3] Westfechted B. Integrated product and process management for engineering design applications [J]. Integrated Computer - Aided Engineering, 1996, 3 (1): 20～35

[4] Kusiak A, Wang J. Decomposition of the design process [J]. Journal of Mechanical Design, 1993, 115 (12): 687～695

Task Arrangement of the IPD Process Plan

Lu Jiansha, Chen Chengpin, Lan Xiuju, Chen Yong, Xie Liewei

(*Institute of Industrial Engineering*, *Zhejiang University of Technology*, *Hangzhou* 310032, *China*)

[**Abstract**] This paper is related to task arrangement of the IPD process plan. The IPD task feature is discussed in regard to coupling and administrative levels. The stratified and distributing IPD process plan strategy is built. The task arrangement modeling and related Hungary algorithms is given with conception of cost matrix, and then related examples are demonstrated.

[**Key words**] IPD; task arrangement; process plan; goal decision