



# 基于分层蚁群算法的人力资源智能优化配置系统研究

(1.东南大学机械工程学院, 南京, 210000; 2.东南大学信息科学与工程学院, 南京, 210096)

**摘要:** 为提高企业人力资源配置的合理性, 设计了一个基于蚁群算法的人力资源智能优化分配系统。本文首先介绍了该系统的框架结构与功能模块, 并研究了利用日志信息修正新任务属性矩阵的方法。随后提出一种分层式蚁群算法用于进行人力资源配置优化, 该算法可简化目标函数及运算过程, 同时可高效的利用反馈信息, 使优化结果更具实效性。最后的实验表明该系统优化结果良好, 具有较强的实用性。

**关键词:** workflow; 蚁群算法; 人力资源; 优化配置

## Human Resource Distribution Smart Optimizing System Using Multi-Layer Ant Colony Optimization

Jiao X<sup>1</sup>, Qi X<sup>1</sup>, Sun XX<sup>1, 2</sup>

(1. School of Mechanical Engineering, Southeast University, Nanjing, 210000;

2. School of Information Engineering, Southeast University, Nanjing, 210096)

**Abstract:** To improve the quality of human resource distribution (HRD) in enterprise, a system of HRD (HRDS) is designed in this paper. Based on the function requirement and the features of HRD, the frame structure and functional module of the system is built in this paper. According to the features of HRD, the state transition rule and pheromone update rule in ant colony optimization (ACO) are appropriately modified, so that the optimizing process can be obviously reduced. Experimental results manifest that this system is accurate, useful and high efficiency.

**Key words:** Workflow; Human resource; Ant colony optimization; Distribution optimizing

制造业内部竞争日益激烈, 如何合理利用自身资源是企业生存与否的一个重要因素。人力资源作为企业的一项重要资源, 其分配的质量将直接影响着企业的运营效果, 从而决定了企业的生存与否。然而, 尽管目前许多企业配套了 ERP、PDM、MES 等信息系统以提升自身的管理与运行效率, 但是在人员与任务匹配方面, 一般仍采用手工方式, 凭借管理者的个人管理经验完成。由于个人经验往往带有主观性、局限性, 难以从全局入手获得最优方案, 这导致人力资源分配成为企业运营的瓶颈。

针对此现状, 近年来许多学者展开了针对人员配置优化的研究, 常见的有基于各类算法(如基于

神经网络增强学习算法<sup>[1]</sup>、multi-objective immune Deadline-based<sup>[2-3]</sup>、locality principle<sup>[4]</sup>和自行设计的 algorithm<sup>[5]</sup>等)或基于各类策略<sup>[6-7]</sup>。但各类优化方法, 大多仅从优化目标入手, 未考虑企业自身分配任务时的固有特点和习惯, 对企业现有的信息管理系统未能加以充分利用。人力资源分配的工作是一种典型的多目标优化问题, 而蚁群算法已被证明是解决该问题的有力工具之一<sup>[8-11]</sup>, 关于蚁群算法在资源优化配置中的应用, 也获得了相应成果<sup>[12-13]</sup>。但是关于人力资源配置方面的研究尚少, 并且在现有研究中, 进行任务分配时所需的参数, 一般由人工指定, 如员工的能力值、任务所需时间等。本文通过将蚁群算法与数据挖掘技术相结合, 将对现有管理平台的数据挖掘结果反馈于分配算法过程, 使得任务分配优化能够在企业固有的任务分配习惯和特点的基础上展开, 所优化的结果更具有实用性。最后的测试结果表明, 该系统具有较高的实用性及效率。

**基金项目:**

**作者简介:**

## 1 系统功能结构

针对企业的信息化管理程度，该系统设计成可作为一个独立系统单独运行，也可作为功能模块，无缝整合入企业现有的信息管理系统。因此设计该系统功能结构如图 1 所示。该系统拥有自身独立的运行功能结构，同时设计了与企业现有管理平台的数据接口。

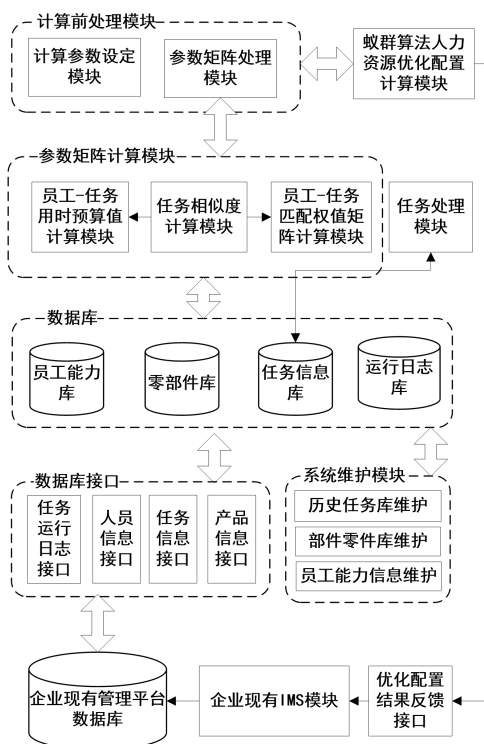


图 1 系统功能结构

数据库：为蚁群优化计算提供参数计算依据；  
系统维护模块：保障系统的正常运行；  
任务处理模块：处理相关的任务信息，使任务信息可用于优化时的参数计算；  
参数矩阵计算模块：利用数据库中现有数据及用户设定，计算出蚁群算法中的所需参数；  
计算前处理模块：根据优化的需求，调整参数矩阵值，同时对蚁群算法进一步设定参数；  
蚁群算法计算模块：根据优化设定参数及参数矩阵进行优化计算；  
优化结果反馈接口及数据库接口：用于与企业现有管理平台的信息交互。

如上功能模块采用模块化设计，因此如果企业现有 IMS 已经包含本系统的部分功能，则对应的功能模块将做简化或是删除即可。一个嵌入 IMS 的该系统的工作流程如图 2 所示。如该系统为独立运行，则应加入任务分解等步骤。

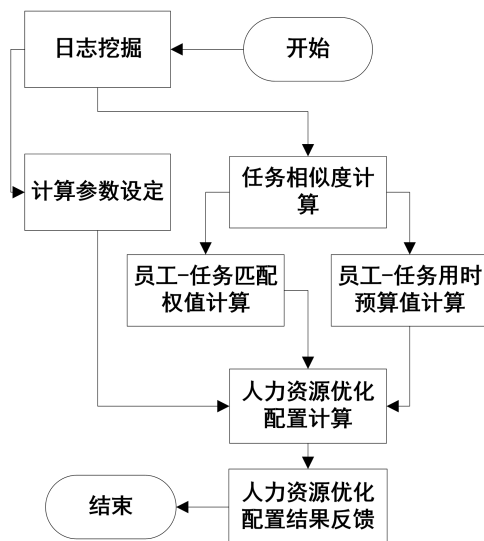


图 2 系统工作流程

## 2 系统关键技术

### 2.1 蚁群优化的信息矩阵预配置

从日志库中获取的相关信息经过处理，以矩阵的形式参与蚁群算法的优化计算，获取此类矩阵的过程称为信息矩阵预配置。主要的相关矩阵有任务相似度矩阵  $S$ 、员工-任务匹配权值矩阵  $W$ 、员工-任务用时估值修正矩阵  $T$ 。

信息矩阵的参考信息来源于对日志库的信息挖掘及对新任务的前期处理（如任务分解、所需成本与时间的估算等）。假定在矩阵计算开始前，如下信息已经得到，同时为能简要提出本文方法的思路，以下均假定任务对象为单一零件，即将任务和零件等同处理：

$l$ : 第  $l$  个待分配任务，其中  $l \in [1, 2, \dots, L]$

$m$ : 日志库中第  $m$  个历史任务， $m \in [1, 2, \dots, M]$

$RL_l$ :  $l$  对应的权限集

$KL_l$ :  $l$  对应的工作岗位类型

$i$ : 第  $i$  个员工， $i \in [1, 2, \dots, I]$

$c_i$ :  $i$  的最近三个月的平均工资

$RI_i$ :  $i$  的权限集

$KI_i$ :  $i$  的工作岗位类型

$et_l$ :  $l$  的估计所需时间

$et'_m$ :  $m$  的估计所需时间

$ht'_{i,m}$ :  $i$  完成  $m$  的实际时间

$\Delta t'_i$ :  $i$  的“惯性时间”，即正式进行被分配任务前的准备时间，主要为当前任务的剩余时间。



$Fi$ :  $i$  曾经完成的任务的集合

$p$ : 任务的第  $p$  项属性

$P$ :  $p$  的可选值集合

$l_p$ : 待分配任务  $l$  的属性  $p$  的值

$m_p$ : 历史任务  $m$  的属性  $p$  的值

$SP_p$ : 属性  $p$  的可选值之间的相似度矩阵

$AP$ : 各属性间的权重系数集合向量

### 2.1.1 任务相似度矩阵 $S$

任务相似度矩阵（以下用  $S$  表示）用于表示待分配任务与历史任务的相似程度。从而使历史任务的相关信息（如完成的时间、成本等）能够为现有任务的对应信息估算提供有价值的参考。因此  $S$  也是其它两个矩阵，即员工-任务匹配权值矩阵  $W$  和员工-任务用时估值修正矩阵  $T$  的参数矩阵。

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,m} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ s_{l,1} & s_{l,2} & \cdots & s_{l,m} \end{bmatrix} \quad (1)$$

式中  $s_{l,m}$  是  $l$  和  $m$  的相似度值。

由于制造业的任务一般以某一个产品或是产品的零部件为工作对象进行，而产品或零部件又由众多子零件所构成。因此当获知了所有重要零部件的相似度后，通过加权计算，即可得出该任务的相似度值。

假设零件的某一属性  $p$  有  $u$  个可能的取值，第  $x$  和第  $y$  个可选值的相似度为  $p_{xy}$ ,  $p_{xy} = p_{yx}$  且  $p_{x,y} \in [0, 1]$ ，则该属性值之间的相似度矩阵如（2）所示：

$$SP_p = \begin{bmatrix} sp_{p,1,1} & sp_{p,1,2} & \cdots & sp_{p,1,u} \\ sp_{p,2,1} & sp_{p,2,2} & \cdots & sp_{p,2,u} \\ \cdots & \cdots & \cdots & \cdots \\ sp_{p,u,1} & sp_{p,u,2} & \cdots & sp_{p,u,u} \end{bmatrix} \quad (2)$$

记零件  $l$  的可选属性集合  $P$  的第  $x$  个值为  $p_x$ ，定义向量  $LPS$  如（3）所示：

$$LPS = \{lps_1, lps_2, \dots, lps_x, \dots, lps_u, \dots\} \\ \text{where } \begin{cases} lps_x = 1, & \text{if } p_x = lp \\ lps_x = 0, & \text{if } p_x \neq lp \end{cases} \quad (3)$$

同理，可得历史任务  $m$  对应的向量  $MPS$ ，则任务  $l$  和任务  $m$  在属性  $P$  方面的相似度  $s_{p,l,m}$  为：

$$s_{p,l,m} = LPS \cdot P \cdot MPS^T \quad (4)$$

故有：

$$s_{l,m} = AP \cdot \{s_{1,l,m}, s_{2,l,m}, \dots, s_{p,l,m}\}^T \quad (5)$$

### 2.1.2 员工-任务匹配权值矩阵 $W$

员工-任务匹配权值矩阵（以下用  $W$  表示）表明员工与某个任务的适合程度，一般受能力、权限等因素影响。

若  $w_{i,l}$  表示员工  $i$  与任务  $l$  的吻合度。则  $W$  可表示为：

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,l} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,l} \\ \cdots & \cdots & \cdots & \cdots \\ w_{i,1} & w_{i,2} & \cdots & w_{i,l} \end{bmatrix} \quad (6)$$

式中：

$$w_{il} = a_{il} \cdot b_{il} \cdot \frac{\sum_{m'=1}^M s_{lm'}}{\sum_{m'=1}^M \sum_{l'=1}^L s_{l'm'}}, \\ \text{where } \begin{cases} a_i = 1, & \text{if } RI_i \subseteq RL_l \\ a_i = 0, & \text{if } RI_i \not\subseteq RL_l \\ b_i = 1, & \text{if } KI_i \subseteq KL_l \\ b_i = 0, & \text{if } KI_i \not\subseteq KL_l \end{cases} \quad (7)$$

### 2.1.3 员工-任务用时估值修正矩阵 $T$

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,l} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,l} \\ \cdots & \cdots & \cdots & \cdots \\ t_{i,1} & t_{i,2} & \cdots & t_{i,l} \end{bmatrix} \quad (8)$$

员工-任务用时估值是进行人员分配优化的重要参考信息。一般企业会对每个任务进行所需时间的预估工作，但是不同的员工的实际完成时间各不相同，而企业无法也没有必要估算每个员工执行每个任务的时间。因此，引入员工-任务用时估值修正矩阵（以下用  $T$  表示）用于存放根据历史日志修正过的当前任务  $l$  的用时估值。

历史数据可以很好的反映各人的能力水平差异，投影到时间方面即为实际所需时间与历史估值时间的偏差。同时，历史任务与当前任务类型越接近，则该历史任务的参考价值越高，故设计修正方

法如式（9）所示：

$$t_{il} = \frac{\sum_{m=1}^M a_m \cdot s_{lm} \cdot ht'_{i,m}}{\sum_{m=1}^M a_m \cdot s_{lm} \cdot et'_m} et'_m \quad (9)$$

$$\text{where } \begin{cases} a_m = 1, \text{ if } a_m \in F_i \\ a_m = 0, \text{ if } a_m \notin F_i \end{cases}$$

与此类似，惯性时间的修正值为：

$$\Delta t_i = \frac{\sum_{m=1}^M a_m \cdot s_{lm} \cdot ht'_{i,m}}{\sum_{m=1}^M a_m \cdot s_{lm} \cdot et'_m} \Delta t'_i \quad (10)$$

## 2.2 基于蚁群算法的人员分配优化

蚁群算法（ant colony optimization, ACO）是一种优秀的启发式算法，最初被设计用于解决 TSP 问题。但是人力资源分配与 TSP 问题有着显著区别，主要体现在如下方面：1.非循环性，在 ACO 中表现为蚂蚁最终不会回到起点；2.非遍历性，在 ACO 中表现为蚂蚁不需要访问到所有的结点；3.可重复性，在 ACO 中表现为结点可重复被选择。

针对如上特点，传统的蚁群算法若直接被应用于人力资源优化分配，会带来目标函数复杂、计算过程繁琐等问题，因此需要根据人力资源分配过程的特点，对传统 ACO 进行一定的改进。关于传统 ACO 的基本函数、参数等定义，可参考<sup>[8-9]</sup>。

### 2.2.1 目标函数

企业对任务完成情况的评判指标一般有三点，即 1.完成时间；2.所需成本；3.完成质量。

目前企业接收的多为定单项目，子任务以达到各项指标均满足为最终目的，不追求超标完成。因此为精简优化过程、提高优化效率，本文视所有任务均可达到质量要求。

同时企业的任务完成成本主要有软性的人力成本及硬性成本如材料成本、生产成本、物流成本等组成。由于针对同一产品，由于指标所限，不同人设计出的产品均大同小异，致使硬性成本浮动很小。因此，成本方面只需考虑人力成本，即员工工资。

在任务分配前，企业一般都会进行项目分解工作。即将项目分解成依次执行的数个的阶段，每个

阶段包含若干个子任务。同一阶段的子任务间无优先顺序。将各阶段的所需时间和成本进行求和，即可得出总项目的所需时间和成本。由于必须执行完上阶段的所有任务才可开始下阶段的工作，所以各阶段的人员优化分配互不干涉。为可简明的表述本文的思想与原理，本文以下涉及的是仅有一个阶段任务的项目。综上所述，以下分别针对时间和成本进行目标函数的建模。

时间目标函数如（10）所示：

$$T = \max(tw_1, tw_2, \dots tw_i) \quad (11)$$

其中：

$$tw_i = \sum_{l=1}^L a_l \cdot t_{il} + \Delta t_i \quad (12)$$

$$\text{where } a_l = \begin{cases} 1, \text{ if } w_l \in AW_i \\ 0, \text{ otherwise} \end{cases}$$

$tw_i$ ：是员工  $i$  在本阶段中执行任务的总时间， $AW_i$  是分配给员工  $i$  执行的任务集

成本目标函数如（11）所示：

$$C = \sum_i c_i \cdot tw_i \quad (13)$$

总目标函数：

$$Aim = \alpha T + \beta C, \text{ where } \alpha, \beta \in [0,1] \quad (14)$$

### 2.2.2 随机状态转移规则

根据人力资源分配的过程特点，即每个任务会被选择，且仅被选择一次，本文将每个任务视为一个母结点，每个任务所对应的候选人集合为该结点

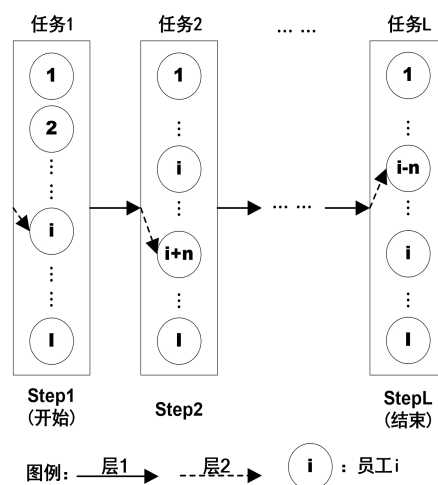


图3 多层蚁群优化法





的子结点。由于在优化计算前并不考虑员工与任务的匹配概率且每个员工可无上限的被重复分配任务，因此每个候选人在每个任务中都有可能被选择到，故每个母结点所包含的候选人集合均相同，即为全体公司员工的集合。考虑到同阶段下的子任务间无优先权，根据概率论的基本概念，母结点的被选择顺序不会影响到最终的优化结果，故 ACO 的优化过程修改成图 3 所示。

为使传统蚁群算法的过程被修改成如图所示，当蚂蚁完成了  $s-1$  步，在第  $s$  步时，处于母结点  $s$  的候选员工集  $WI_s$  中的员工  $i$  被选中的概率为：

$$P_{s-1,s}^k = \begin{cases} \frac{\tau_{s,i}^a \cdot \eta_{s,i}^b}{\sum_{r \in WI_s} \tau_{s,r}^a \cdot \eta_{s,r}^b}, & \text{if } i \in WI_s \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

式中  $\tau_{s,i}^a$  是任务  $s$  第二层的路径：任务  $s$  到员工  $i$

这一路径的信息素。 $\eta_{s,i}^b$  是该路径所对应的启发因子，即能见度。 $a \in [0, 1]$  和  $b \in [0, 1]$  用于调整信息素和能见度之间的权重。式中没有涉及到一般 ACO 优化均需采用的 tabu 表。

能见度从一定程度上决定着蚂蚁发现该子结点的概率。因此可将由历史日志库所挖掘到的信息应用于此，从而使日志信息可反馈控制当前的人力资源优化匹配过程。

任务  $s$  的第二层结点  $i$ ，即员工  $i$  的能见度  $\eta_{s,i}^b$  为：

$$\eta_{s,i}^b = \frac{\delta}{TotalT_{i,s}} \cdot \frac{1}{t_{i,s} \cdot c_i} \quad (16)$$

其中：

$$TotalT_{i,s} = \sum_{l=1}^L a_l \cdot t_{i,l}, a_l = \begin{cases} 1, & \text{if } w_l \in LW_i \\ 0, & \text{others} \end{cases} \quad (17)$$

式中： $LW_i$  是在第  $s$  步时，员工  $i$  已经被分配的任务集合。 $\delta > 0$  是用于调节时间和成本权重比的参数。

### 2.2.3 信息素更新规则

若信息素始终处于积累状态，那么随着优化的不停循环，信息素将会增加到淹没能见度信息的地步。因此必须在每次循环结束后，对信息素进行升华。为增加收敛速度，本文采用全局更新法，即路径  $s-i$  上，时间  $t$  到  $t+1$  时，信息素更新规则如下<sup>[8]</sup>：

$$\tau_{s,i}(t+1) = (1-\rho)\tau_{s,i}(t) + \Delta\tau_{s,i}(t) \quad (18)$$

将日志信息反馈于人力资源优化，得到信息素增加函数为：

$$\Delta\tau_{s,i}(t) = \frac{Q}{AT_k \cdot AC_k} \quad (19)$$

式中： $AT_k$  和  $AC_k$  是蚂蚁  $k$  在  $t$  时的循环中所需的总时间和总成本，求解方法可由 (11)、(13) 而得。 $Q$  是用于控制收敛速度的参数。

## 3 实验结果

在应用时，实际工作时间会受诸多因素干扰，如病假、工作失误等等，因为本文为验证多层蚁群算法在优化中的实际效果，并未使用现实中的案例，而是利用程序，按一定的算法随机生成模拟现实的所需信息，如任务历史日志、员工属性、任务属性等。将这些信息，通过穷举法，可获得一个目标函数的理论最佳解。再利用多层蚁群算法进行人力资源的优化分配，将优化结果与最佳值进行对比，即可验证出本文方法的优劣。

本文模拟了一个只有 1 个阶段、8 和 9 个子任务的项目，有 5 位各有特点的员工待分配。优化的

表 18 任务时实验数据

方法	任务 1 到 8 各自 对应的员工	目标函 数值	所需时间 (天)	优化时间 (秒)
EM	2,1,4,2,3,5,1,5	132.4	24	74
2000	2,1,3,2,3,4,1,5	132.6	25	8
1000	2,1,3,2,3,4,1,5	132.6	25	4
300	2,1,4,2,3,4,1,5	135.4	27	1
100	2,1,4,2,5,4,3,5	140.6	28	<1

注：表中 EM 代表穷举法，100、300、1000、2000 分别表示循环 100 次、300 次、1000 次、2000 次的多层蚁群算法的结果。

表 29 任务时实验数据

方法	任务 1 到 9 各自 对应的员工	目标函 数值	所需时间 (天)	优化时间 (秒)
EM	2,1,4,3,3,5,1,5,2	159.2	29	554
2000	2,1,4,3,3,4,1,5,2	159.9	30	10
300	2,3,4,1,3,4,1,5,2	162.5	32	2



代码为 C#, 运行的硬件平台为 CPU: P4D, 内存: 2G (双通道), 945 主板, 7200 转 SATA 硬盘。

在目标函数 (14) 中,  $\alpha = \beta = 0.5$ , 在转移概率函数 (15) 中,  $a=b=1$ , 信息素增加函数中  $Q=100$ 。优化结果如表 1、表 2 所示。

实验结果表明蚁群算法可以在较短的时间内, 获得近似于最佳值的结果, 效率极高。同时, 仅增加一个子任务时, 穷举算法的所需时间提高了 7 倍多, 而对应的蚁群算法仅增加了 25%。由于实际工作中子任务的数量通常远远大于本次实验, 因此显而易见蚁群算法更适用于实际应用。

## 参考文献

- [1] 张毅, 罗元, 郑太雄等. 移动机器人技术及其应用[M]. 北京: 电子工业出版社, 2007.
- [2] Ishiguro, H.Y., M. Tsuji, et al. Pattern analysis and machine intelligence[J]. IEEE Transaction on, 1992, 14(2): 257-262.
- [3] 廖晓辉, 沈大中, 王东署. 智能轮椅的研究现状与关键技术分析[J]. 制造业自动化, 2008, 30(4): 1-6.
- [4] R.S.Rao, K.Conn, S.H.Jung, et al. Human robot interaction: application to smart wheelchairs[C]. International Conference on Robotics & Automation, Washington, DC: 7803-7272, 2002: 3583-3588.
- [5] Richard Simpson, Edmund LoPresti, Steve Hayashi, et al. The smart wheelchair component system[J]. Journal of Rehabilitation Research & Development, 2004, 41(3B): 429-442.
- [6] 章小兵. 基于视觉和超声波传感器的地面移动机器人定位与路径规划研究[D]. 南京: 东南大学, 2006.
- [7] 赵海文, 岳宏, 张雅丽等. 移动机器人运动目标跟踪系统设计[J]. 微计算机信息, 2007, 23(9-2): 249-251.
- [8] 荣命哲, 姜建勇, 王小华. 永磁式接触器动触头动作特性仿真分析与试验研究[J]. 中国电机工程学报, 2005, 25(1): 109-113.
- [9] Fang Shuhua, Lin Heyun, et al. Magnetic field analysis and dynamic characteristic prediction of ac permanent-magnet contactor[J]. IEEE Trans. Magnetics, 2009, 45(7): 2990-2995.
- [10] 房淑华, 林鹤云, 蔡彬等. 永磁接触器磁场有限元分析及控制单元设计[J]. 中国电机工程学报, 2006, 26(22): 162-166.
- [11] 毛万搏, 林鹤云, 汪先兵等. 永磁接触器操动机构多目标动态优化设计[J]. 东南大学学报(自然科学版), 2009, 39(2): 287-292.
- [12] 汪先兵, 林鹤云, 房淑华等. 永磁接触器位移分段 PWM 控制及吸合过程动态特性分析[J]. 中国电机工程学报, 2010, 30(3): 113-118.
- [13] Chien T C. Applying closing phase-angle control technique in bounce reduction of AC permanent magnet contactor[J]. Hindawi Publishing Corporation Active and Passive Electronic Components, 2009, (2009): 1-12.