

基于蚁群算法的飞行器管理系统任务调度

孙大伟, 董新民, 郭创
(空军工程大学航空航天工程学院, 西安 710038)

摘要: 为解决飞行器管理系统的任务调度问题, 提出了一种基于蚁群算法的多处理机任务调度方法, 考虑任务个数、任务等级、任务负载、任务与飞行状态关系等4个因素构建了单机风险系数函数; 以单机风险系数均衡为目的设计了任务调度目标函数, 给出了蚁群算法求解飞行器管理系统任务分配问题的流程。仿真得到一种较优的调度方案, 验证了所提方法的有效性。

关键词: 飞行器管理系统; 任务调度; 蚁群算法; 组合优化

中图分类号: V247.1 **文献标志码:** A **文章编号:** 1671-637X(2014)08-0083-04

Task Scheduling of Vehicle Management System Based on ACO Algorithm

SUN Da-wei, DONG Xin-min, GUO Chuang
(Engineering College of Aeronautics and Astronautics, Air Force Engineering University, Xi'an 710038, China)

Abstract: A method based on ant colony optimization (ACO) was put forward for solving the problem of multiprocessor task scheduling in Vehicle Management System(VMS). Taking task number, task grade, task load, and the relation between task and flight status into consideration, the single processor risk coefficient function was designed. To balance the risk coefficients among the processors, the objective function of task scheduling was established. Then, the process of ACO for solving VMS task scheduling problem was given. A suboptimal scheduling scheme was obtained through simulation, which verified the effectiveness of proposed method.

Key words: Vehicle Management System (VMS); task scheduling; Ant Colony Optimization (ACO); combinatorial optimization

0 引言

综合控管是飞行控制发展的必然趋势, 飞行器管理系统(Vehicle Management System, VMS)将飞机安全飞行必不可少的功能, 包括飞行控制、推进控制、公共设备管理(UMS)以及电源管理等综合起来, 在物理上和功能上对整个飞机进行统一控制与管理, 实现子系统协调、功能合理分配以及资源共享, 优化了飞行器性能, 减轻了驾驶员负担^[1-3]。

全透明的VMS体系结构^[4]由多台VMS处理机以及VMS子系统组成, 各子系统的控制管理工作根据不同的准则划分为多个任务分布在多台处理机节点执行, 以保持飞机VMS性能最佳。任务调度是VMS的核心技

术^[1], 直接关系到分布式管理系统的性能。调度的核心是资源的分配, 即如何分配资源以保证每个任务在约束条件内完成^[5]。VMS的任务调度则是由VMS处理机对系统的资源进行分配, 在满足资源占用要求和实时性要求的前提下, 通过网络总线, 动态地或静态地调度处理机节点间的任务, 使得各处理机负载均衡, 保证每个任务满足它们的时间约束, 同时响应外部请求。当处理机或任务发生故障时, 系统有故障容错和重构功能。

1 问题描述

VMS体系结构如图1所示。

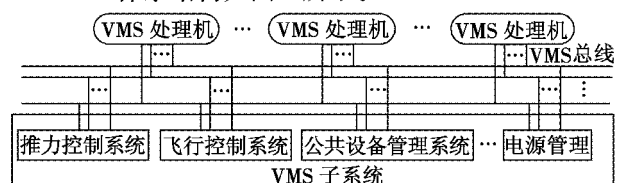


图1 VMS体系结构

Fig.1 VMS architecture

收稿日期: 2013-11-18 修回日期: 2014-01-07

基金项目: 国家自然科学基金(61304120)

作者简介: 孙大伟(1991—), 男, 山西运城人, 硕士生, 研究方向为飞行器导航、制导与控制。

典型的 VMS 任务调度问题可以描述如下:现有 n 个任务与 m 个处理机节点,任务包括周期任务与非周期任务,本文只研究周期任务的调度问题。任务在处理机节点间的调度需满足以下约束条件:1)所有任务按开始时间排序,开始时间早的排在前面;2)相同开始时间的任务按照任务优先级执行,优先级高的先执行;3)每个处理机节点某一时刻只能执行一个任务;4)某一处理机节点上所有任务的负载总和不能超过该处理机的处理能力;5)某一处理机节点上所有任务的排序要保证所有周期任务能够按时完成。

所谓调度,就是在满足上述约束条件的前提下,采用一定的调度算法,分配所有的 n 个任务到 m 个处理机节点,并使其满足调度的目标函数,其实质是组合优化问题。主要包括以下 4 个方面。

调度对象:表示所有调度任务的集合,用 $T\{t_1, t_2, \dots, t_n\}$ 表示。

调度目标:由 $\{M, J\}$ 表示, M 代表各处理机节点的集合, J 是通过 VMS 总线连接的各处理机的分布式控制结构。

调度算法:表示从调度对象至调度目标的方法。

目标函数:表示整个任务调度所要达到的目标,用 D 表示。

2 任务调度目标函数

考虑 n 个任务与 m 个处理机节点的任务调度问题,各处理机节点用 $M_j (1 \leq j \leq m)$ 表示,节点 M_j 上执行 $n_j (1 \leq n_j \leq n)$ 个任务,任务用 $T(t_1, t_2, \dots, t_{n_j})$ 表示。

从 VMS 系统的可靠性以及安全性角度出发,VMS 任务调度的目标选择为使得每一个处理机节点的失效对整机的影响最小。当某一处理机节点故障时,为了使 VMS 系统正常运行,须将该节点的任务转移至其余正常工作的处理机节点,否则将使整机丧失部分功能,其过程是一个任务再分配过程。因此进行多处理机的任务调度,从节点可能故障角度考虑,如何使其余正常工作的处理机节点接受再分配的任务,使任务再分配过程风险降至最低,是需要关心的问题。

定义单机风险系数 R_j :单机故障时,会导致其上任务不能完成,将可能对飞机飞行安全造成威胁。单机风险系数表示故障的影响程度,风险系数越大,故障对整个系统影响就越大。根据飞行器管理系统特点,某一处理机节点的单机风险系数与以下 4 个因素有关。

1) 节点任务数。

节点任务数是指处理机节点 M_j 上已被分配的任务数量,用 n_j 表示。 n_j 的大小代表了该节点故障时失效任务的多少,当 n_j 较大时,该节点的任务再分配给

其余节点的分配负担也相应较大,风险也会较大。定义 A_j 为任务数量风险系数, A_j 正比于 n_j ,反映节点任务数量对单机风险系数的贡献程度。

2) 节点任务等级。

节点任务等级是指处理机节点 M_j 上所有任务的重要程度等级,用 G_i 表示。按照任务失效对飞行安全的影响程度不同,将任务划分为 3 个等级:三级任务为任务失效将对飞机飞行安全造成直接威胁的任务;二级任务为任务失效将不会威胁飞行安全,但会影响飞机部分功能;一级任务为任务失效只会对飞机部分不影响飞行安全的辅助功能造成影响的任务。定义 B_i 为任务等级风险系数, B_i 正比于 G_i ,反映节点任务等级对单机风险系数的贡献程度。

3) 节点任务负载。

节点任务负载是指处理机 M_j 上所有任务对处理机资源的需求,反映了任务对处理机能力的占用程度,用 L_i 表示。某一任务的负载 L_i 可以用该任务在处理机上的执行时间 E_i 的某一倍数与该任务周期 P_i 之比计算, $L_i = l \cdot E_i / P_i$,其中 l 为常值倍数。

对于处理机节点而言,接收任务时既要保证自己能够完成执行已有任务,又要保证其剩余处理能力足够处理被接收的任务。当某个任务负载较大的处理机的节点发生故障时,其上任务的分配要求其处理机有较高的剩余处理能力,若其余处理机不能满足要求,将拒绝接受该节点分配的任务。所以故障处理机任务负载越大,分配调度难度越高。定义 C_i 为任务负载风险系数, C_i 正比于 L_i ,反映了节点任务负载对单机风险系数的贡献程度。

4) 飞机飞行状态。

飞机的飞行状态根据其飞行时刻不同而有所改变,在滑跑、爬升、平飞、降落等不同的飞行状态,VMS 子系统的工作情况亦不尽相同,某些仅在特有飞行状态下较为重要的任务在其他飞行状态失效时所带来的风险相应较小。将飞机的飞行状态分为空中状态和地面状态,相应地将各飞行状态下的任务划分为两级,定义 G_{Fi} 为任务的当前飞行状态重要度等级:二级任务为与当前飞行状态有着直接联系的任务,任务失效将会影响当前飞行状态;一级任务为与当前飞行状态不直接联系的任务,任务失效仅造成轻微影响或不造成影响。定义 S_i 为飞行状态风险系数, S_i 正比于 G_{Fi} ,反映任务与当前飞行状态重要度等级对单机风险系数的贡献程度。

综合以上分析,可将单机风险系数表示为

$$R_j = a \cdot A_j + \sum_{i=1}^{n_j} (b \cdot B_i + c \cdot C_i + d \cdot S_i) \quad (1)$$

式中: $A_j = n_j$; $B_i = e \cdot G_i$; $C_i = L_i = l \cdot E_i / P_i$; $S_i = f \cdot G_{Fi}$ 。
 R_j 为处理机节点 M_j ($1 \leq j \leq m$) 的单机风险系数, a 、 b 、 c 、 d 为各因素加权系数, e 、 f 、 l 均为常值。

单机风险系数反映的是单个处理机节点故障对 VMS 系统正常工作的影响程度。现从整个 VMS 系统多处理机节点间多任务调度的全局出发, 要使单处理机节点故障对整个 VMS 系统造成的影响最小。

对于整个系统而言, 在某一飞行状态的任意时刻, 所有处理机任务总数 n 是固定的, 相应所有任务的任务等级风险系数之和 $\sum B_i$ 与任务负载风险系数之和 $\sum C_i$ 也固定, 所以整个系统总的风险系数是一定的。这就意味着若某一处理机节点单机风险系数较小, 必定会存在另外的单个或几个处理机风险系数较大, 若这些风险系数较大的处理机节点发生故障, 将对整个系统正常运行造成很大影响, 从综合控管系统安全性需求出发, 不希望出现这种情况, 所以采用各处理机节点平摊总风险系数的策略来降低整个系统的风险, 只有当各处理机单机风险系数均衡时, 单机失效对整个系统影响最小。因此定义任务分配目标函数为

$$D = \min \sum_{j=1}^m \left[\left(R_j - \frac{1}{m} \sum_{j=1}^m R_j \right)^2 \right] \quad (2)$$

3 调度算法设计

一般来说, 组合优化问题是一个 NP 完全问题, 现存的主要解决办法多采用启发式的进化算法, 蚁群算法由于其特有的优点, 已被证明在解决复杂组合优化问题上有独特的优势, 如经典的旅行商问题 (TSP)、二次分配问题 (QAP)、车间任务调度问题 (JSP) 等^[6-7]。

3.1 蚁群算法介绍

蚁群算法^[8]是模拟真实蚁群觅食过程寻求最短路径的原理, 由意大利学者 M. Dorigo 等人首先提出。算法通过候选解组织群体的进化过程来寻求最优解。进化过程包括两个阶段: 一是适应阶段, 各候选解根据积累的信息不断调整自身结构; 二是协作阶段, 各候选解间通过信息交流, 以便产生性能更好的解。

在蚁群算法中, 每只蚂蚁根据问题依赖的准则, 从被选的初始状态出发, 建立一个可行解或是解的一个组成部分。当大量蚂蚁不断地从蚁巢通往食物时, 相同时间内相对较短路径上通过的蚂蚁较多, 该路径上累积的信息素强度也较大, 后来蚂蚁选择该路径的概率也相对较大, 相互协作的一群蚂蚁很容易找到从蚁巢到食物源的最短路径, 因此蚁群算法具有并行计算、正反馈、群体智能的特点。

3.2 构建解

首先依据任务调度问题的约束条件将需要调度的

任务进行排列, 组成一定的任务矢量集合 $T\{t_1 \ t_2 \ \dots \ t_n\}$, 蚁群算法的任务就是将这些任务分配至不同的处理机节点。在构建路径之前, 先将总共 K 只蚂蚁随机地分布在各处理机上, 然后在路径构建的每一步中, 蚂蚁 k 按照任务矢量的顺序移动, 从而将不同的任务转移至相应的处理机节点, 每个蚂蚁转移任务的过程构成系统的一个解。蚂蚁 k 把一个任务 t_i 从当前处理机 M_i 转移至另外的处理机 M_j 的转移概率为

$$P_{(i,j)}(k) = \frac{\tau_{(i,j)}^\alpha \cdot \eta_{(i,j)}(k)^\beta}{\sum_{j=1}^m \tau_{(i,j)}^\alpha \cdot \eta_{(i,j)}(k)^\beta} \quad (3)$$

式中: $k \in [1, K]$, 为蚂蚁的序号, K 为蚂蚁个数; $\tau_{(i,j)}$ 为当前任务 t_i 与处理机节点 M_j 之间的信息素, 所有的信息素都用 τ_0 进行初始化, τ_0 是一个很小的值; α 和 β 是两个权值参数, 分别决定信息素和启发式信息的相对影响力; $\eta_{(i,j)}(k)$ 表示的是启发式信息, 在本问题中, $\eta_{(i,j)}(k)$ 设置为任务 t_i 从当前处理机 M_i 转移至处理机 M_j 的启发值, 即

$$\eta_{(i,j)}(k) = \begin{cases} R_i - R_j, & R_i > R_j \\ 0, & R_i \leq R_j \end{cases} \quad (4)$$

只有当前处理机 M_i 的单机风险系数 R_i 大于处理机 M_j 的风险系数 R_j 时, 才对蚂蚁转移任务有启发效果, 再由 $P_{(i,j)}(k)$ 定义可知, 该启发值越大, 转移概率越大, 对蚂蚁的指示作用也就越大。

3.3 信息素局部更新规则

蚂蚁 k 位于处理机节点 M_i , 根据转移概率选择出当前任务 t_i 转移的下一个处理机节点 M_j 后, 使用式 (5) 进行局部信息素更新。作用在于减少蚂蚁经过的路径的信息素, 从而使其他蚂蚁选中该边的概率相对减少, 增加探索尚未经过的路径的机会, 增强了算法的“勘探”能力, 从而有效避免算法进入停滞状态。

$$\tau_{(i,j)} = (1 - \rho) \cdot \tau_{(i,j)} + \rho \cdot \tau_0 \quad (5)$$

式中, $0 < \rho < 1$, 是信息素的挥发系数。

3.4 信息素全局更新规则

当所有的蚂蚁将任务矢量的所有任务转移到相应的处理机节点后, 用式 (6)、式 (7) 对信息素进行全局更新。从信息素全局更新规则可以看出, 在一次循环的所有蚂蚁中, 只有那些目标函数最优的蚂蚁所经过路径的信息素才被增强, 其余蚂蚁经过的路径的信息素不会进行全局调整。信息素全局更新规则的使用使算法的寻优过程更具有指导性, 全局信息素的更新规则为

$$\tau_{(i,j)} = (1 - \lambda) \cdot \tau_{(i,j)} + \lambda \cdot \Delta\tau_{(i,j)}, (i,j) \in \text{global best} \quad (6)$$

$$\Delta\tau_{(i,j)} = \begin{cases} Q/D_{\text{best}}, \forall (i,j) \in \text{global best tour} \\ 0, & \text{其他} \end{cases} \quad (7)$$

其中: $0 < \lambda < 1$ 是信息素衰减系数; D_{\min} 是所有蚂蚁本次循环的目标函数最优值; Q 为常数。

3.5 算法描述

用蚁群算法求解 VMS 任务调度问题的算法流程如图 2 所示。

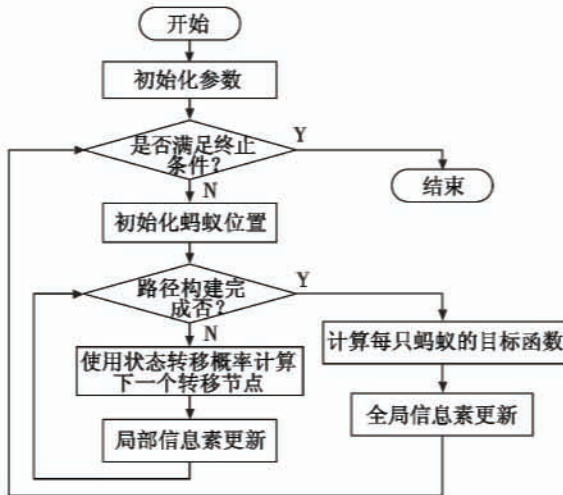


图2 蚁群算法流程图
Fig.2 Flow chart of ACO

具体步骤如下所述:

- 1) 初始化算法的各项参数,包括 $\tau_0, \alpha, \beta, \rho$, 以及蚂蚁个数 K , 最大迭代次数 L_{\max} 等;
- 2) 初始化蚂蚁位置,将所有蚂蚁随机放置在各处理机节点;
- 3) 使用状态转移概率对蚂蚁的转移节点进行计算,并用式(5)进行信息素的局部更新;
- 4) 所有蚂蚁都进行完一次循环后,每只蚂蚁就代表一个解,计算出所有蚂蚁中目标函数最优蚂蚁,并用式(6)、式(7)进行信息素全局更新;
- 5) 判断是否满足结束条件,满足则输出最优目标函数,否则继续迭代。

4 仿真分析

假定 10 个典型 VMS 子任务在 4 台 VMS 处理机上的调度问题进行仿真实验,任务列表如表 1 所示。任务等级依据对飞行安全的影响划分,例如飞控子系统中舵面控制任务、公共设备管理系统中起落架收放任务、发动机加力控制任务等对飞行安全影响较大的为三级任务,飞控系统自动驾驶任务、屏显任务等不影响飞行安全但会使飞机丧失部分功能的为二级任务,环控系统中座舱温度显示任务等仅对飞机部分辅助功能造成影响的为一级任务。假设飞机飞行状态为空中状态,飞行状态任务等级按照任务与当前飞行状态之间的联系划分,例如舵面控制任务、发动机控制任务、电

传控制任务等与空中状态有直接联系的任务为二级任务,防滑刹车系统任务、减速伞控制任务等与空中状态无直接联系的任务为一级任务。

表 1 任务列表

Table 1 Task list

| 任务编号 | 任务执行时间/ms | 任务周期/ms | 任务等级 | 飞行状态任务等级 |
|------|-----------|---------|------|----------|
| 1 | 5 | 50 | 3 | 1 |
| 2 | 6 | 50 | 3 | 1 |
| 3 | 10 | 80 | 2 | 2 |
| 4 | 7 | 100 | 3 | 1 |
| 5 | 6 | 120 | 2 | 2 |
| 6 | 15 | 125 | 2 | 2 |
| 7 | 10 | 125 | 2 | 1 |
| 8 | 10 | 150 | 1 | 1 |
| 9 | 12 | 180 | 1 | 2 |
| 10 | 5 | 200 | 1 | 1 |

取加权系数 a, b, c, d 为 1, $l=2, e=0.33, f=0.8$ 。蚁群算法具体参数设置如下:全局更新系数 $\lambda=0.2$, 局部更新系数 $\rho=0.2, \alpha=3, \beta=2, L_{\max}=60, K=20, Q=2$ 。得到蚁群算法目标函数收敛曲线如图 3 所示。

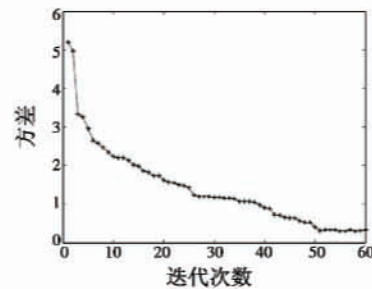


图3 目标函数寻优过程

Fig.3 The optimization of objective function

表 2 给出了 10 个子任务分配给 4 台处理机目标函数最优时的具体分配方案以及各处理机单项风险系数的具体比较,此时 $D_{\min}=0.3005$ 。由于所分析的问题规模较小,单个任务的风险系数对单机风险系数的贡献较大,所以表 2 中调度方案所得到的各处理机的单机风险系数之间存在一定差异,是在满足目标函数最小前提下得到的相对均衡的分配结果。

表 2 调度方案

Table 2 Scheduling scheme

| 处理器编号 | 分配的任务 | 任务等级风险系数 | 任务负载风险系数 | 飞行状态风险系数 | 单机风险系数 |
|-------|--------|----------|----------|----------|--------|
| 1 | 2,6 | 1.6667 | 0.4533 | 1 | 5.12 |
| 2 | 3,5 | 1.3334 | 0.35 | 1.6 | 5.2834 |
| 3 | 1,7,8 | 2 | 0.4934 | 0.6 | 6.0934 |
| 4 | 4,9,10 | 1.6667 | 0.3233 | 1.2 | 6.19 |

5 结束语

多处理机正常运行是 VMS 正常工作的保证,其任
(下转第 97 页)

- [8] LI Y, BAI B D, ZHANG Y N. Improved particle swarm optimization algorithm for fuzzy multi-class SVM[J]. Journal of Systems Engineering and Electronics, 2010, 21(3): 509-513.
- [9] VAN DEN BERGH F. An analysis of particle swarm optimizers[D]. Pretoria; University of Pretoria, 2001.
- [10] DING W P, WANG J D, GUAN Z J. Cooperative extended rough attribute reduction algorithm based on improved PSO[J]. Journal of Systems Engineering and Electronics, 2011, 23(1): 160-166.
- [11] STIMSON G W. Introduction to airborne radar[M]. 2nd ed. Raleigh, NC, USA: SciTech Publishing, Inc., 1998.
- [12] GRIFFITHS H D, BAKER C J. Passive coherent location radar systems, part 1: Performance prediction[J]. IEE Proc. Radar Sonar Navig, 2005, 150(3): 153-159.
- [13] 路志伟, 李明, 季晓光. 基于传感器协同技术的雷达搜索空域研究[J]. 航空计算技术, 2006, 36(5): 5-8.
- LU Z W, LI M, JI X G. Research on radar searching volume based on multi-sensor cooperation technology[J]. Aeronautical Computing Technique, 2006, 36(5): 5-8.

(上接第 62 页)

- [14] TICHVASKY P, MURAVCHIK C H, NEHORAI A. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering[J]. IEEE Transactions on Signal Processing, 1999, 46(5): 1386-1395.
- [15] 杜继永, 张凤鸣. 基于改进粒子群算法的多UCAV任务分配仿真研究[J]. 系统仿真学报, 2013, 25(4): 650-655.
- DU J Y, ZHANG F M. Simulation study on task assignment for multi-UCAV based PSO[J]. Journal of System Simulation, 2013, 25(4): 650-655.
- [16] 赵远东, 方正华. 带有权重函数学习因子的粒子群算法[J]. 计算机应用, 2013, 33(8): 2265-2268.
- ZHAO Y D, FANG Z H. PSO algorithm with weight function's learning factor[J]. Journal of Computer Applications, 2013, 33(8): 2265-2268.
- [17] 高雷卓, 刘旭旺. 基于混沌的弹性粒子群全局优化算法[J]. 控制与决策, 2009, 24(10): 1545-1548.
- GAO L F, LIU X W. Resilient particle swarm global optimization algorithm based on chaos[J]. Control and Decision, 2009, 24(10): 1545-1548.
- [18] 杨雪榕, 梁加红, 陈凌, 等. 多邻域改进粒子群算法[J]. 系统工程与电子技术, 2010, 32(11): 2453-2458.
- YANG X R, LIANG J H, CHEN L, et al. Multi-neighborhood improved PSO algorithm[J]. Systems Engineering and Electronics, 2010, 32(11): 2453-2458.

(上接第 86 页)

务调度是 VMS 的关键技术之一。为了量化处理机故障的风险, 本文综合考虑任务个数、任务等级、任务负载、任务与当前飞行状态关系 4 个因素, 提出了单机风险系数概念, 并以单机风险系数均衡为目标函数, 采用蚁群算法在组合优化方面的独特优势, 设计了算法流程, 获得了较为满意的离线性能和结果。

参考文献

- [1] 高丽, 吴文海, 曲志刚. 飞行器管理系统研究综述[J]. 飞行力学, 2010, 28(3): 1-4.
- GAO L, WU W H, QU Z G. Survey of research for vehicle management systems[J]. Flight Dynamics, 2010, 28(3): 1-4.
- [2] 罗海明, 谢剑斌, 陆志肖. 机电系统综合化控制和管理[J]. 直升机技术, 2010, 1: 62-67.
- LUO H M, XIE J B, LU Z X. Integrated control and management for utility management system[J]. Helicopter Technique, 2010, 1: 62-67.
- [3] HAYRE A, DULL T, MEYN F. The ATF YF-23 vehicle management system[C]//Aerospace Design Conference, Irvine, CA, 1992, doi:10.2514/6.1992-1076.
- [4] MOIR I, SEABRIDGE A G. Management of utility system in the experimental aircraft programmer[J]. Aerospace, 1996, 9: 28-35.
- [5] 左伟, 冯金富, 潘博. 分布式开放实时系统系统调度策略研究[J]. 电光与控制, 2011, 18(9): 35-39.
- ZUO W, FENG J F, PAN B. Research on scheduling scheme for distributed open real-time system[J]. Electronics Optics & Control, 2011, 18(9): 35-39.
- [6] 邓玉芬, 向凤红. 蚁群算法在组合优化中的应用[J]. 电子测量技术, 2007, 30(1): 32-35.
- DENG Y F, XIANG F H. Application of ant colony algorithm in combination and optimization[J]. Electronic Measurement Technology, 2007, 30(1): 32-35.
- [7] 段海滨, 王道波, 于秀芬. 蚁群算法的研究现状及其展望[J]. 中国工程科学, 2007, 9(2): 98-102.
- DUAN H B, WANG D B, YU X F. Research status and prospect of ant colony algorithm[J]. Engineering Science, 2007, 9(2): 98-102.
- [8] 段海滨. 蚁群算法及其应用[M]. 北京: 科学出版社, 2005.
- DUAN H B. Ant colony algorithm and application[M]. Beijing: Science Press, 2005.