

一种多机器人任务规划算法及其系统实现

余伶俐 焦继乐 蔡自兴

(中南大学信息科学与工程学院 长沙 410083)

摘要 针对多机器人任务规划问题,提出了一种蚁群集中式规划方法,建立了任务分配和路由规划的蚁群算法描述模型,并利用局部搜索策略改进了蚁群算法分配效果,实现了多机器人集中任务规划系统。利用该系统平台,进行了大量的实验分析。结果表明,蚁群算法能有效解决多机器人任务规划问题,为多机器人协作机制提供了新思路。

关键词 多机器人,任务规划,蚁群算法,集中分配,路由规划

中图分类号 TP301.6 **文献标识码** A

Multi-robot Mission Planning Algorithm and its System Implementation

YU Ling-li JIAO Ji-le CAI Zi-xing

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract We proposed a centralized ant colony planning approach for multi-robot mission planning problem. The task allocation model and route planning model of ant colony algorithm description were built, meanwhile, local search strategy was utilized to improve ant colony algorithm assignment effect. At last, we achieved a centralized multi-robot mission planning system for experiments analysis. The results show that ant colony algorithm can solve the multi-robot mission planning problem effectively, and provide a new idea for multi-robot coordination mechanisms.

Keywords Multi-robot, Mission planning, Ant colony algorithm, Centralized allocation, Route planning

多机器人任务规划以各种模型为基础,根据规划指标为多机器人设计出协同的执行方案,使其整体效能最优或近似最优。任务规划的好坏在很大程度上决定了多机器人执行任务的效能、代价以及故障发生率等。因此,任务规划的研究对提高多机器人系统的效率有重要意义。

任务规划可由任务分配与路由规划两部分构成。目前,基于市场竞拍机制的任务分配方法^[1,2],是将任务分配视为交易过程,必须保证同一机器人的各个交易之间相互独立且线性,必须有一个初始解,但大规模问题的初始解却难以确定。网络流量优化方法^[3]将多机器人任务分配看作一个商业供需网络物流优化问题,但其鲁棒性有待改进。基于割图任务分配方法^[4]建立起点至任务的有向图,将任务分配问题看作有向收益图划分子图的过程,但其计算量大。近年来,随着群集机器人分配技术的发展,文献^[5]提出了一种响应分配阈值模型,并出现了利用免疫系统^[6]解决相对动态的任务分配问题,多机器人任务分配系统的实现也得到了较大的发展^[7,8]。本文设计了集中式任务分配系统,其可降低大量的通信代价,使得整体方案近似最优解。利用蚁群算法集中分配,解决了中大规模的任务规划问题,克服了初始解设定困难和计算量大的不足,其分配结果优于 C 均值集中分配^[9]和迭代自组织算法(Iterative Self-organizing Data Analysis Technology Algorithm, ISODATA)^[10]集中分配。路由规划算法

发展也较为迅速。路由规划指根据任务分配的结果,规划完成任务的行进路由(执行次序),以满足指标最优性要求。Voronoi^[11]和 A^* ^[12]是经典启发式搜索算法求解路径规划,近年来针对不同的应用对象均有较大改进。文献^[13]提出了一种新的插入算法求解路由调度,但这些算法随着进化算法^[14]与仿生群算法^[15]的发展受到了一定的挑战。蚁群算法^[16]作为一种分布式寻优算法,搜索能力强,最早成功应用于规划问题,所以本文采用蚁群算法求解此类组合优化问题。首先针对集中任务分配与路由规划建立了蚁群算法的数学描述,并定义了理想任务与探测任务不同的目标函数,提出了一种局部搜索策略来改进任务分配效果,最后实现了集中任务规划系统。基于此系统完成了各种算法的大量对比实验,结果表明蚁群算法能有效解决集中任务规划问题。

1 多机器人集中式任务分配的蚁群算法描述

1.1 理想环境下集中式分配算法问题描述

假设机器人任务地图已知,任务系统不随时间变化,各机器人能完成所分配任务。理想环境中忽略任务执行时间, N 个任务按照给定目标分配给 K 个机器人,使得系统分配结果最优。

a) 构造蚂蚁结构:每只蚂蚁表示一种可能的分配结果,搜索前为各蚂蚁设置一个长度为任务数 N 的空解集 S ,解集中

到稿日期:2009-07-08 返修日期:2009-10-12 本文受 NSFC 重大研究计划资助项目(90820302),国家博士点基金项目(200805330005)资助。

余伶俐(1983-),女,博士生,助教,主要研究方向为多机器人规划,E-mail:llyu@csu.edu.cn;焦继乐(1988-),女,硕士生,主要研究方向为先进机器人控制;蔡自兴(1938-),男,教授,博士生导师,主要研究方向为人工智能、机器人学。

的第 i 个位置对应于第 i 个任务所分配的机器人号。

b) 构造信息素矩阵: 集中式任务分配问题中, 信息素是一个在迭代过程中不断更新的 $N \times K$ 矩阵。元素 τ_{ij} 代表任务 i 分配到第 j 个机器人的信息素值, 初始化为常数 τ_0 。

c) 构造目标函数: 根据集中式分配原则, 计算各机器人所分配任务到虚拟分配中心的距离之和的总和和最小值作为目标函数, 如式(1)所示。或定义各机器人所分配任务到虚拟分配中心的最大距离最小作为分配目标, 如式(2)所示。定义虚拟分配中心如式(3)所示, 其中 w_{ij} 的含义如式(4)所示。

$$\min F(w, c) = \min \left(\sum_{j=1}^K \sum_{i=1}^{N_j} \sum_{v=1}^n w_{ij} \|x_{iv} - c_{jv}\|^2 \right) \quad (1)$$

$$\min F(w, c) = \min \left[\max \left(\sum_{i=1}^{N_1} \sum_{v=1}^n w_{i1} \|x_{iv} - c_{1v}\|^2, \sum_{i=1}^{N_2} \sum_{v=1}^n w_{i2} \|x_{iv} - c_{2v}\|^2, \dots, \sum_{i=1}^{N_j} \sum_{v=1}^n w_{ij} \|x_{iv} - c_{jv}\|^2 \right) \right] \quad (2)$$

$$c_{jv} = \frac{\sum_{i=1}^{N_j} w_{ij} x_{iv}}{\sum_{i=1}^{N_j} w_{ij}} \quad (j = 1, \dots, K; v = 1, \dots, n) \quad (3)$$

$$w_{ij} = \begin{cases} 1, & \text{若样品 } i \text{ 属于 } j \text{ 类} \\ 0, & \text{否则} \end{cases} \quad (j = 1, \dots, K; i = 1, \dots, N) \quad (4)$$

d) 更新蚁群: 预先定义阈值概率 q_0 ($0 < q_0 < 1$), 通过信息素的间接通信实现 N 个任务分配于 K 个机器人的近似划分, 系统产生一个随机数 q 。若随机数小于 q_0 , 按最大信息素值 τ_{ij} 选择机器人; 否则, 每个任务的 τ_{ij} 按式(5)转换为选择概率 p_{ij} 。利用轮盘赌法进行选择, 既利用了已有知识, 也开发了新解空间。

$$p_{ij} = \frac{\tau_{ij}}{\sum_{i=1}^M \tau_{ij}} \quad (j = 1, \dots, M) \quad (5)$$

e) 信息素矩阵更新: 利用蚂蚁的解对信息素矩阵进行更新。更新采用式(6), 式(7), 其中 ρ ($0 < \rho < 1$) 为信息素蒸发系数, $\tau_{ij}(t)$ 为 t 时刻任务 i 分配给机器人 j 的信息素值。Q 为品质因数。F_s 为目标函数值。至此, 一次迭代结束。迭代直至达到最大迭代次数, 返回最优集中式分配结果。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{s=1}^l \Delta\tau_{ij}^s \quad (6)$$

$$\Delta\tau_{ij}^s = \begin{cases} Q/F_s, & \text{样品 } i \text{ 属于 } j \text{ 类} \\ 0, & \text{否则} \end{cases} \quad (7)$$

1.2 集中式探测任务分配算法描述

探测任务分配需充分考虑任务执行时间, 这与理想环境下的任务分配有所不同, 可用最大时间最小, 式(8)作为目标函数。蚁群更新和信息素矩阵的更新方法与 1.1 节相同。但式(7)中 F_s 表示蚂蚁 s 的最大时间最小目标值 T 。本文定义机器人速度 vel (取每秒 4 个距离单位) 和任务执行时间 t_{exei} ($i = 1, \dots, N$)。设置参数时需使任务执行时间和路由行驶时间数量级相当。

$$\min T(w, c) = \min \left[\max \left[\sum_{i=1}^{N_1} \sum_{v=1}^n w_{i1} \|x_{iv} - c_{1v}\|^2 / vel + \sum_{i=1}^N t_{exei}, \dots, \sum_{i=1}^{N_j} \sum_{v=1}^n w_{ij} \|x_{iv} - c_{jv}\|^2 / vel + \sum_{i=1}^N t_{exei} \right] \right] \quad (8)$$

1.3 局部搜索策略改进集中式分配蚁群算法

R 只蚂蚁每次迭代结束后, 加入局部搜索, 进一步提高集中式分配的质量。把所有的解按照目标函数进行升序排列, 采用变换操作对具有较小目标函数值的前 L 个 (本文 L 取 2) 解进行局部搜索操作, 其实现步骤如图 1 所示, 其中 pls 表示随机数阈值。

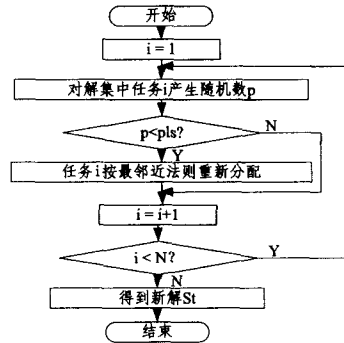


图 1 局部搜索流程图

1.4 改进蚁群算法解决任务分配问题的实现流程

根据 1.1 节假设以及算法分析, 蚁群集中式分配算法流程如图 2 所示。

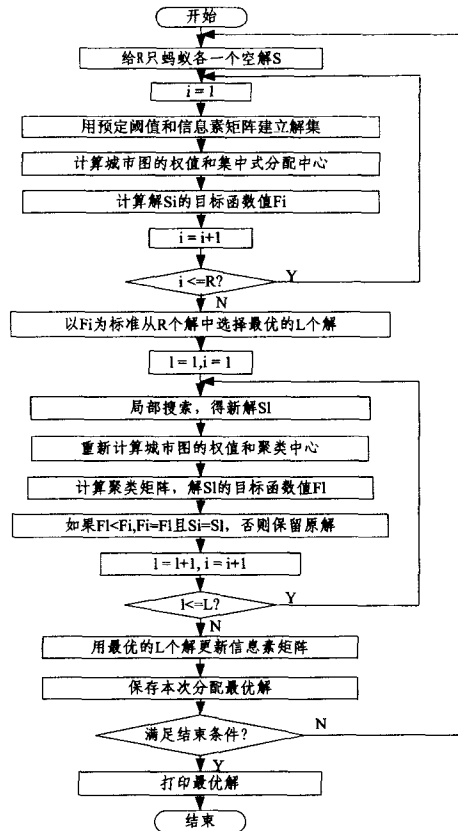


图 2 任务分配流程图

2 解决机器人路由规划的蚁群算法描述

2.1 蚁群算法解决路由规划问题描述

假设机器人任务地图已知, 任务系统不随时间变化。利用 R 只蚂蚁找到一条机器人遍历 N 个任务的最短路由, 如图 3 所示。

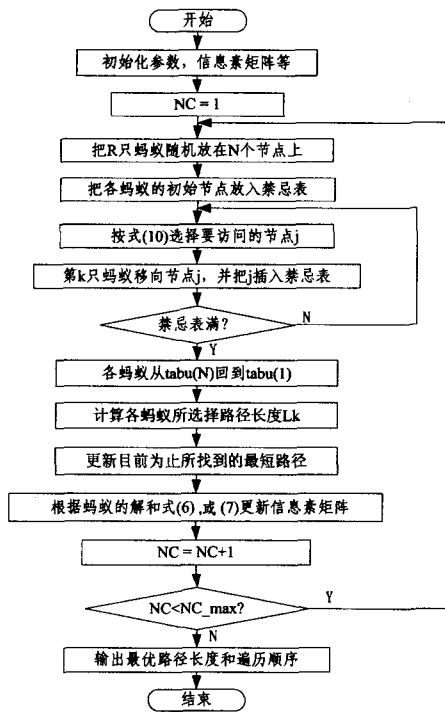


图3 路由规划流程图

2.2 蚁群算法解决路由规划算法描述

a) 构造蚂蚁结构: 蚁群算法解决路由规划问题中, 每只蚂蚁表示一种可能的遍历方式。

b) 构造信息素矩阵: 路由规划问题中, 信息素是在迭代过程中不断更新的 $N \times N$ 矩阵。 τ_{ij} 代表任务 i 到任务 j 路由上的信息素值, 初始化为常数 τ_0 。

c) 构造目标函数: 设蚂蚁初始解 $S = \{s_1, s_2, \dots, s_N\}$, s_i 表示任务完成次序。遍历各任务所经路由之和最小作为目标函数, 其数学模型为式(9), 其中 n 表示任务特征个数, x_{iv} 为第 i 个任务的第 v 个特征。

d) 更新蚁群: 在每一次蚁群更新中, 蚂蚁将通过信息素的间接通信重新遍历 N 个任务。方法如下: 设第 $t-1$ 次迭代后蚂蚁 k 位于任务 i 选择下一步要执行的任务 j , 选择概率为式(10)。其中, $allowed_k = \{N - tabu_k\}$, $tabu_k$ 为禁忌表, 存放蚂蚁已经执行过的任务, 以保证任务不重复执行。 α 和 β 是表示信息启发式因子及期望启发式因子相关重要性的参数。 τ_{ij} 为任务 i 到任务 j 路由上信息素浓度, η_{ij} 为任务 i 到任务 j 的启发式信息, d_{ij} 为任务 i 到任务 j 的路由长度。

e) 信息素矩阵更新: 采用类似式(6), 式(7)和蚂蚁的解对信息素矩阵进行更新。其中 ρ ($0 < \rho < 1$) 为信息素蒸发系数, F_s 为蚂蚁 s 的目标函数值, Q 为品质因数。

至此, 一次迭代结束。继续迭代, 直至最大迭代次数, 返回最优路由长度及遍历次序的规划结果。

$$\min F = \min \left(\sum_{i=1}^{N-1} \sum_{j=i+1}^N \|x_{iv} - x_{(i+1)v}\|^2 + \sum_{v=1}^n \|x_{Nv} - x_{1v}\|^2 \right) \quad (9)$$

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}, & \text{如果 } j \in allowed_k \\ 0, & \text{否则} \end{cases} \quad (10)$$

$$\eta_{ij} = 1/d_{ij} \quad (11)$$

3 多机器人任务规划实验分析与系统实现

为使结果具有参照性和可比性, 实验选用 TSPLIB 库中

3 组标准数据 (eil51, eil76, eil101) 进行测试。图 4 为中南大学智能机器人实验室的 MORCS-2 机器人团队实物图。表 1 为蚁群算法中各参数的设置。



图4 MORCS-2 机器人团队

表 1(a) 算法参数(1)

q_0	α	β	ρ	Q	MAX_NC	蚂蚁数
0.9	1.0	5.0	0.1	100	200	等于任务数

表 1(b) 算法参数设置(2)

任务号	t_{exei}	任务号	t_{exei}	任务号	t_{exei}
eil51		eil76		eil101	
1-20	1	1-20	1	1-20	1
21-40	2	21-40	2	21-40	2
41-51	3	41-76	3	41-101	3

3.1 集中式任务规划实验分析

实验将改进蚁群算法分配算法与集中式 C 均值分配算法和 ISODATA 进行对比, 数据结果统计如表 2 至表 5 所列。表中数据均为 10 次实验的平均值。在理想环境下, 解决多机器人集中任务规划时本文算法效果优于 C_mean 和 ISODATA。针对探测任务规划, 蚁群算法的效果也优于 C_mean 和 ISODATA。但在表 5 中, 3 个机器人执行 76 个任务时, 蚁群算法所得全局最小时间为 309.84, 最大时间最小值为 119.22; 而 ISODATA 算法得到的全局最小时间为 318.10, 最大时间最小值为 115.66。即 ISODATA 算法所得最大时间最小值优于蚁群算法。这是因为 ISODATA 是一种专业的集中式分配算法, 其平均分配的效果有时接近蚁群分配水平。但由于 ISODATA 算法分配的类数由算法决定而非预定的, 因此它不适合用在机器人数目已知的任务规划问题中。这同时也说明了蚁群算法还有改进空间。在以后的研究中, 可以对蚁群算法做进一步的改进, 比如引入多 agent 联盟机制分配任务, 进而求得更优的解。

表 2 理想任务规划全局路由最小 3 种算法实验数据结果

算法	蚁群算法			C_mean			ISODATA		
	3	4	5	3	4	5	3	4	5
eil51	447.07	452.78	460.89	512.48	568.35	548.46	—	—	—
eil76	567.79	579.27	585.28	685.99	705.67	651.54	600.35	—	—
eil101	676.51	692.96	714.60	817.74	828.93	822.40	698.89	—	—

表 3 理想任务规划最大路由最小 3 种算法实验数据结果

算法	蚁群算法			C_mean			ISODATA		
	3	4	5	3	4	5	3	4	5
eil51	157.36	134.73	114.46	215.46	178.22	172.03	—	—	—
eil76	211.93	178.72	157.27	300.34	275.26	231.27	222.63	—	—
eil101	263.21	222.07	228.83	345.43	239.01	225.40	307.48	—	—

表 4 针对探测任务规划全局时间最小 3 种算法实验数据结果

算法	蚁群算法			C_mean			ISODATA		
	3	4	5	3	4	5	3	4	5
eil51	183.89	185.08	201.85	221.12	235.09	230.07	—	—	—
eil76	309.84	312.43	315.1	339.75	344.44	330.89	318.10	—	—
eil101	411.30	415.54	420.17	447.44	450.16	448.71	417.83	—	—

表 5 针对探测任务规划最大时间最小 3 种算法实验数据结果

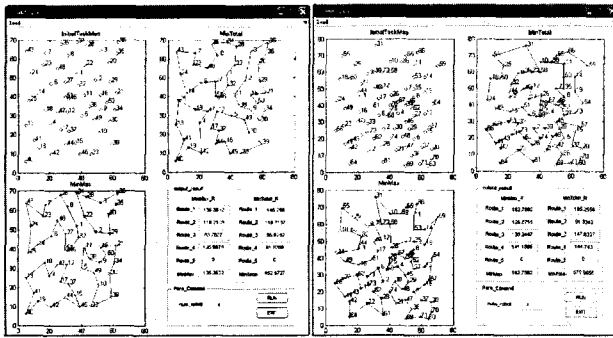
算法	蚁群算法	C_mean	ISODATA
----	------	--------	---------

机器人数	3	4	5	3	4	5	3	4	5
eil51	64.23	54.38	48.76	94.87	70.92	76.97	—	—	—
eil76	119.22	101.09	88.12	132.22	113.83	107.82	115.66	—	—
eil101	153.22	132.01	121.52	178.36	136.56	136.61	191.82	—	—

3.2 多机器人任务规划系统的实现

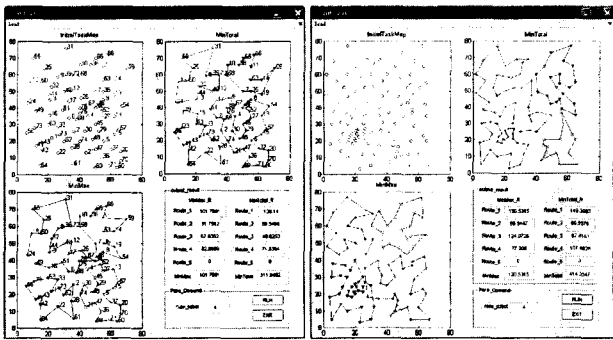
多机器人任务规划系统实现是改进型蚁群算法解决集中式任务分配、路由规划和人机界面的组合。菜单 load 调初始任务地图,并显示在 Initial_map 坐标轴中,在 num_robot 后面文本框中输入参数,即完成任务所使用的机器人数量。

点击按钮 RUN,程序开始运行。程序运行完毕,在坐标轴 MinTotal 和坐标轴 MinMax 上分别显示按总代价最小和最大代价最小的任务规划结果。针对理想任务规划,在 result 的文本框里显示总最小路由和最大值最小路由。针对探测式任务规划显示相应各时间量,点击 EXIT 退出程序。图 5 为理想条件下蚁群算法针对地图 eil51, eil76, 4 个机器人任务规划的结果,图 6 为蚁群算法规划探测任务的结果,图 7 和图 8 分别为 4 个机器人用 C 均值与 ISODATA 算法执行 76 个探测任务的规划结果。



(a) 4 个机器人执行 51 个任务 (b) 4 个机器人执行 76 个任务

图 5 理想条件下蚁群算法任务规划结果



(a) 4 个机器人执行 76 个任务 (b) 4 个机器人执行 101 个任务

图 6 蚁群算法探测任务规划结果

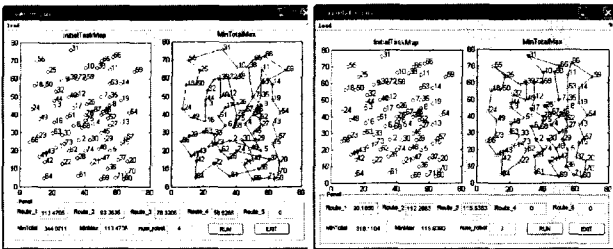


图 7 探测任务规划 C_mean 算法 4 个机器人执行 76 个任务

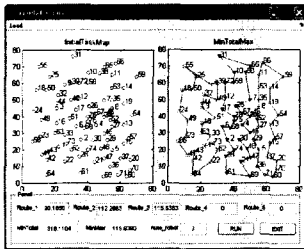


图 8 探测任务规划 ISODATA 3 个机器人执行 76 个任务

结束语 利用蚁群算法解决多机器人任务分配与路由规划问题,针对性地建立了各种问题的蚁群解结构,并与其他集中分配算法进行了对比和分析。实验结果表明蚁群算法能较好地解决多机器人任务规划问题。本文对蚁群算法的改进是一种尝试,仍有进一步提升空间,例如可引入多 agent 联盟机制进一步优化蚁群算法。另外,本文针对静态环境而言,集中式分配可满足要求,但对于动态环境的任务规划还需进一步从算法体系结构上进行研究。

参考文献

- [1] Gerkey B, Mataric M. Sold! Auction Methods for Multi-robot Coordination [J]. IEEE Trans. on Robotics and Automation, 2002, 18(5): 758-768
- [2] Gao Ping-an, Cai Zi-xing. Multi-robot task allocation for exploration [J]. Journal of Central South University: Science and Technology, 2006, 13(5): 548-551
- [3] Chandler P R, Rasmussen S R. Task Allocation for Wide Area Search Munitions via Iterative Network Flow[C]//Proceedings of AIAA Guidance, Navigation and Control Conference. 2002: 345-351
- [4] Chandler P R, Pachter M. Complexity in UAV Cooperative Control [C]//Proceedings of the American Control Conference. Anchorage, AK, 2002: 5-10
- [5] Yang Yong-ming, Zhou Chang-jiu, Tian Yan-tao. Swarm Robots Task Allocation Based on Response Threshold Model[C]//Proceedings of the 4th International Conference on Autonomous Robots and Agents. Wellington, New Zealand, 2009
- [6] Gao Yun-yuan, Luo Zhi-zeng. Dynamic Task Allocation Method Based on Immune System for Cooperative Robots[C]//Proceedings of the 7th World Congress on Intelligent Control and Automation. Chongqing, 2008
- [7] Ulam P, Endo Y, Wagner A, et al. Integrated Mission Specification and Task Allocation for Robot Teams, Design and Implementation[C]//2007 IEEE International Conference on Robotics and Automation. Roma, Italy, 2007
- [8] Sarel-Talay S, Balch T R, Erdogan N. Multiple Traveling Robot Problem: A Solution Based on Dynamic Task Selection and Robust Execution[J]. IEEE/ASME Trans. on Mechatronics, 2009, 14(2): 198-206
- [9] 王海军, 邓羽, 王丽, 等. 基于数据场的 C 均值聚类方法研究[J]. 武汉大学学报: 信息科学版, 2009, 34(5): 626-629
- [10] Wang Ying, Li Ren-Wang, Li Bin, et al. Research on an Ant Colony ISODATA Algorithm for Clustering Analysis in Real Time Computer Simulation[C]//Digital Media and Its Application in Museum & Heritages, Second Workshop. 2007: 223-229
- [11] Li Qing-quan, Zeng Zhe. A Voronoi-based Hierarchical Graph Model of Road Network for Route Planning. Intelligent Transportation Systems[C]// 11th International IEEE Conference. 2008: 599-604
- [12] Tulum K, Durak U, Yder S K. Situation aware UAV mission route planning[C]// Aerospace Conference. 2009: 1-12
- [13] 孔媛, 唐加福, 董钢, 等. 插入算法求接送顾客到机场的车辆调度问题[J]. 控制理论与应用, 2009, 26(1): 92-96
- [14] Nanayakkara S C, Srinivasan D, Lai Wei Lup, et al. Genetic Algorithm based route planner for large urban street networks[C]//Evolutionary Computation IEEE Congress. 2007: 4469-4474
- [15] Wang Zhen-hua, Zhang Wei-guo, Shi Jing-ping, et al. UAV route planning using multiobjective ant Colony system[C]// Cybernetics and Intelligent Systems, IEEE Conference. 2008: 797-800
- [16] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesmen Problem[J]. IEEE Transactions on Evolutionary computation, 1997, 1(1): 53-65