

改进蚁群优化算法和多邻域下降搜索的混合启发式算法求解该问题。

人工蜂群算法 ABC(Artificial Bee Colony Algorithm)^[9~12]由 Karaboga D 在 2005 年提出,算法模仿蜂群的觅食行为,是一种基于群体智能的非数值优化计算方法。算法最初用来解决函数的优化问题,由于其控制参数少、易于实现、计算简洁等优点,目前已被广泛运用于很多实际问题,例如数字过滤器^[13]、最小生成树^[14]、机器人路径规划^[15]等。但是,目前国内外学者将人工蜂群算法用于求解离散问题的研究还比较少,Singh A^[14]利用人工蜂群算法找出具有叶子约束的最小生成树;胡中华等^[15]将人工蜂群算法应用于机器人路径规划问题中;胡中华等^[16]运用人工蜂群算法求解旅行商问题;孙晓雅等^[17]利用人工蜂群算法对任务指派问题进行了求解。

本文在已有研究的基础之上,充分考虑车辆路径问题解的离散性特点,给出了食物源的自然数编码方法,并且采用邻域倒位操作的方法生成候选食物源,使人工蜂群算法可以用来求解车辆路径问题等组合优化问题。经过若干实例的仿真实验,并与其他算法进行比较,表明了本文算法在求解车辆路径问题中的可行性和有效性。

2 车辆路径问题的描述与数学模型

车辆路径问题一般可描述如下:中心仓库使用 K 辆具有相同负载能力的车辆向分仓库配送货物,求在满足货运的需求下,使车辆的行程路线最短。该问题有如下限制:(1)每个分仓库的配送任务仅由一辆车服务且只能服务一次;(2)所有的车辆必须从中心仓库出发,最后再回到中心仓库;(3)每条配送路线上分仓库的需求总和不能超过车辆的负载;(4)每个分仓库需要配送的数量不能超过车辆的负载。下面给出该问题的数学模型:

记 $G = (V, E)$ 为赋权图, $V = \{0, 1, 2, \dots, n\}$ 为顶点集,其中 0 表示中心仓库, $V' = \{1, 2, \dots, n\}$ 表示分仓库集, E 为边集,各顶点间的距离为 $c_{ij} (i, j \in V)$,每辆车的负载为 Q ,各仓库的需求为 $d_i (i \in V)$,且定义变量:

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 由分仓库 } i \text{ 驶向分仓库 } j \\ 0, & \text{其它} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{分仓库 } i \text{ 的货运任务由车辆 } k \text{ 完成} \\ 0, & \text{否则} \end{cases}$$

得到数学模型^[18]如下:

$$\min Z = \sum_{i \in V} \sum_{j \in V} \sum_{k=1}^K c_{ij} x_{ijk} \quad (1)$$

$$\text{s. t.} \quad \sum_{i \in V} d_i y_{ik} \leq Q, \forall k = 1, 2, \dots, K \quad (2)$$

$$\sum_{k=1}^K y_{ik} = \begin{cases} 1, & \forall i \in V' \\ K, & i = 0 \end{cases} \quad (3)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \forall i \in V, k = 1, 2, \dots, K \quad (4)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \forall S \subseteq V',$$

$$|S| \geq 2, k = 1, 2, \dots, K,$$

$$x_{ijk}, y_{ik} \in \{0, 1\}, \forall i, j \in V, k = 1, 2, \dots, K \quad (5)$$

在上述数学模型中,式(2)表示每条配送路线上分仓库的需求总和不能超过车辆的负载;式(3)表示配送任务由 K 辆车完成,且每个分仓库的配送任务仅由一辆车完成; S 为顶点集的任一子集合, $|S|$ 为集合 S 中所含图 G 的顶点个数。于是,对每个可能的 S ,式(5)意味着一组约束,式(4)和式(5)保证了可行回路的形成。

3 人工蜂群算法

在人工蜂群算法中,人工蜂群主要由引领蜂(Employed Bees)、跟随蜂(Onlookers)和侦察蜂(Scouts)三个部分组成。人工蜂群算法在求解优化问题时,食物源代表优化问题的一个可能解,蜂群采蜜(食物源)的过程也就是搜寻优化问题最优解的过程。食物源的优劣取决于优化问题的适应值(函数值),适应值高的食物源较优。人工蜂群算法中解的个数(SN)等于引领蜂或跟随蜂的个数。用 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 表示第 i 个食物源($i = 1, 2, \dots, SN$, D 为搜索空间的维数)。首先,人工蜂群算法随机产生 SN 个解(食物源),然后蜂群对所有的食物源进行循环搜索,循环次数为 MCN 。引领蜂首先在食物源的邻域生成一个候选食物源,并比较候选食物源与先前食物源的优劣,如果候选食物源的适应值优于先前食物源的适应值,则用候选食物源代替先前的食物源,否则保持先前的食物源不变。结束之后,引领蜂回到舞蹈区把食物源优劣的信息通过跳摇摆舞传达给跟随蜂,跟随蜂根据所得到的信息按照一定概率选择食物源,适应值越高的食物源被选择的概率越大。跟随蜂选中食物源后,也在食物源的邻域生成一个候选食物源,并比较候选食物源与选中食物源的优劣,保留较优的

食物源。人工蜂群算法就是通过上述反复循环来最终找到最优解的。当某个蜜蜂个体经过 $limit$ 次循环食物源没有更新时,个体放弃该食物源变成侦察蜂,寻找新的食物源。

引领蜂和跟随蜂根据式(6)在食物源的邻域生成一个候选食物源:

$$v_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{kj}) \tag{6}$$

其中, v_{ij} 是生成的候选食物源, $k \in \{1, 2, \dots, SN\}$, $j \in \{1, 2, \dots, D\}$, 这两个数都是随机选取的, 但 $k \neq i$, r_{ij} 是 $[-1, 1]$ 上均匀分布的随机数, 它控制 x_{ij} 邻域的生成范围。随着迭代次数的增加, $(x_{ij} - x_{kj})$ 之间的距离缩小, 搜索空间也缩小, 即搜索步长缩小, 动态地调整步长有助于提高算法精度, 并最终获得最优解。

人工蜂群算法中跟随蜂根据食物源的收益率来选择到哪个食物源采蜜。收益率通过适应值来表示, 食物源选择的概率为:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{7}$$

其中 fit_i 为第 i 个食物源的收益率。

人工蜂群算法中, 某个蜜蜂个体经过 $limit$ 次循环食物源没有更新时, 个体放弃该食物源变成侦察蜂, 新的食物源由式(8)生成:

$$x_i^j = x_{min}^j + rand(0, 1)(x_{max}^j - x_{min}^j) \tag{8}$$

其中, x_i^j 为新的食物源的第 j 维分量, $j \in \{1, 2, \dots, D\}$, $rand(0, 1)$ 为 $[0, 1]$ 的随机数, x_{min}^j, x_{max}^j 分别为第 j 维变量的最小值和最大值。

4 求解车辆路径问题的人工蜂群算法

4.1 构造食物源编码

在前面的描述中可以看出, 人工蜂群算法中食物源编码是实数编码, 而车辆路径问题是离散问题。文献[5]求解车辆路径问题时采用的是实数编码, 然后再按实数大小排序, 得到一条自然数编码的可行线路, 方法虽然可行, 但效率不高, 为提高效率, 本文对食物源的编码直接采用自然数编码。车辆路径问题的一条可行线路可编成长度为 $k + m$ 的食物源 $(0, t_{11}, t_{12}, \dots, t_{1u}, 0, t_{21}, t_{22}, \dots, t_{2v}, 0, \dots, 0, t_{m1}, t_{m2}, \dots, t_{mw})$ 。这个食物源可以解释为第一辆车从中心仓库 0 出发到达分仓库 $t_{11}, t_{12}, \dots, t_{1u}$, 最后返回中心仓库; 第二辆车从中心仓库 0 出发到达分仓库 $t_{21}, t_{22}, \dots, t_{2v}$, 最后返回中心仓库; 以此类推, 第 m 辆车从中心仓库 0 出发到达分仓库 t_{m1} ,

t_{m2}, \dots, t_{mw} , 最后返回中心仓库。例如, 设有八个分仓库, 三辆车, 若食物源 $x = 01203450678$, 则表示的行车路线为: 第一辆车: 中心仓库出发到达分仓库 1、2, 最后返回中心仓库; 第二辆车: 中心仓库出发到达分仓库 3、4、5, 最后返回中心仓库; 第三辆车: 中心仓库出发到达分仓库 6、7、8, 最后返回中心仓库。

文献[1]中的编码, 虽然也采用了自然数编码, 但编码的意义不同, 文献[1]中的个体为 $[1, k \cdot m]$ 的一个互不重复的自然数。本文编码与其相比, 缩小了搜索空间, 提高了收敛速度, 在迭代次数较少时就可以收敛到最优解。

4.2 生成候选食物源

食物源采用实数编码时, 引领蜂和跟随蜂通过式(6)来生成候选食物源。由于车辆路径问题的食物源采用自然数编码, 再按照式(6)来生成候选食物源显然是不可行的, 为保证生成的候选食物源仍然是自然数编码, 我们采用邻域倒位操作的方式来生成候选食物源。以八个分仓库、三辆车的车辆路径问题实例来说明生成候选食物源的方法。如图 1 所示, 当前食物源为 x_i , 随机选取的相邻食物源为 x_k , 它们倒位前编码如图 1a 所示。邻域倒位的过程为: 随机选择的位置 d (x_i 和 x_k 在该位置的值不能为 0), 此处 $d = 2$, 箭头①指向 x_k 的第二位“4”, 箭头②指向 x_i 的第二位“1”, 然后在 x_i 中寻找“4”的位置, 这里是第六位, 箭头③表示将 x_i 的第二位“1”与第六位“4”之间的数字进行倒位操作, 由倒位前的 12034 得到倒位后的 43021, 流程如图 1a 中箭头①②③的流向所示。这样倒位后生成的候选食物源如图 1b 所示。由于生成的候选食物源对食物源 x_i 的变动不是很大, 仍然会具有 x_i 中很多良好的特性; 同时, 由于进行邻域倒位的位置是随机选取的, 这可以增加食物源的多样性, 防止算法陷入早熟收敛, 增强了人工蜂群算法在求解车辆路径问题时的全局搜索性能。

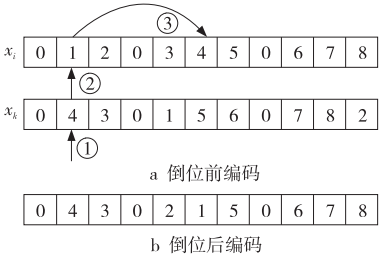


Figure 1 Neighborhood inversion to produce a candidate food source

图 1 邻域倒位操作方法生成候选食物源

4.3 适应值函数

在车辆路径问题的数学模型中,式(2)要求每条配送路线上分仓库的需求总和不能超过车辆的负载。本文通过目标函数(1)及约束条件(2)加罚函数法来获得适应值函数:

$$\min Z = \sum_i \sum_j \sum_k c_{ij} x_{ijk} + R \cdot \sum_k \max(\sum_i d_i y_{ki} - Q, 0) \tag{9}$$

其中, R 作为惩罚系数,是一个很大的正数。这样,不可行解会被赋予极大的适应值,在人工蜂群算法迭代中会被淘汰掉。

5 仿真实验及分析

为验证所提算法的有效性,运用 VC++ 语言编程实现了上述算法,并对几个国内外常用的车辆路径问题基准测试实例进行了仿真实验。

5.1 实例 1

采用文献[2,3,5]都选用的一个实例。该实例有八个分仓库和一个中心仓库,中心仓库有两辆负载为 8t 的车可以用于配送货物。中心仓库和各分仓库之间的距离及各分仓库对中心仓库的需求量如表 1 所示(其中 0 表示中心仓库),要求合理安排行驶路线,使车辆的行程最短。

Table 1 Distance and demand of the central warehouse and branch warehouses
表 1 中心仓库与分仓库之间距离及需求量

	0	1	2	3	4	5	6	7	8
0	0	4	6	7.5	9	20	10	16	8
1	4	0	6.5	4	10	5	7.5	11	10
2	6	6.5	0	7.5	10	10	7.5	7.5	7.5
3	7.5	4	7.5	0	10	5	9	9	15
4	9	10	10	10	0	10	7.5	7.5	10
5	20	5	10	5	10	0	7	9	7.5
6	10	7.5	7.5	9	7.5	7	0	7	10
7	16	11	7.5	9	7.5	9	7	0	10
8	8	10	7.5	15	10	7.5	10	10	0
需求量	1	2	1	2	1	4	2	2	

实验中,本文算法的参数设置如下:种群规模为 60,迭代次数为 50(与文献[2,3,5]的设置相同)。文献[2]分别采用标准遗传算法(GA)和双种群遗传算法(DPGA)对该实例进行了 20 次计算,其中标准遗传算法的平均值为 73.25,找到已知最优解的百分率为 0,双种群遗传算法的平均值是 69.575,找到已知最优解的百分率为 20%;文献

[3]采用混合遗传算法(HGA)对该问题进行了 20 次计算,平均值是 67.875,找到已知最优解的百分率为 75%;文献[5]用改进微粒群算法(MPSO)在 20 次计算中得到的平均值是 68.375,找到已知最优解的百分率为 45%;本文算法在 20 次计算中得到的平均值是 67.65,找到已知最优解的百分率为 90%。本文算法的求解结果与上述算法的求解结果如表 2 所示。

Table 2 Results of example 1
表 2 实例 1 的计算结果

次数	GA	DPGA	HGA	MPSO	本文算法
1	74.0	70.0	67.5	67.5	67.5
2	75.0	69.5	67.5	69.5	69.0
3	71.5	67.5	67.5	67.5	67.5
4	72.0	71.0	67.5	67.5	67.5
5	73.5	69.0	69.0	69.0	67.5
6	75.0	70.5	67.5	69.0	67.5
7	73.0	72.0	69.0	69.0	67.5
8	72.5	67.5	69.0	67.5	67.5
9	75.5	71.5	67.5	69.0	67.5
10	73.5	69.0	67.5	67.5	67.5
11	72.0	67.5	67.5	67.5	67.5
12	69.0	69.0	67.5	69.0	67.5
13	73.0	71.0	67.5	69.0	67.5
14	75.5	70.0	69.0	67.5	69.0
15	72.0	67.5	67.5	69.5	67.5
16	73.0	70.5	67.5	69.0	67.5
17	75.0	69.0	69.0	67.5	67.5
18	73.5	69.5	67.5	69.0	67.5
19	71.5	71.0	67.5	69.0	67.5
20	75.0	69.0	67.5	67.5	67.5

由表 2 可见,在种群规模和迭代次数相当的情况下,本文算法的求解结果明显优于标准遗传算法、双种群遗传算法、混合遗传算法和改进微粒群算法。本文算法求得最优解线路详细调度为:第一辆车:中心仓库出发到达分仓库 4、7、6,最后返回中心仓库;第二辆车:中心仓库出发到达分仓库 1、3、5、8、2,最后返回中心仓库。

5.2 实例 2

上述实例规模较小,虽然在文献[5]中提及了较大规模的实例,但是实验数据比较简单,不能有效验证算法对较大规模车辆路径问题的适用性。因此,我们以文献[3]采用的国际上公认的车辆路径问题数据库中两个较大规模的基准测试实例进行了计算,实例数据可以从 <http://branchandcut.org/VRP/data/> 下载。本文算法的求解结果与文献[3]混合遗传算法的求解结果如表 3 所示。

Table 3 Results of the E-n22-k4 and E-n33-k4

表 3 E-n22-k4 和 E-n33-k4 的计算结果

实例名 (已知最优解)	算法	种群 规模	迭代 次数	1	2	3	4	5	6	7	8	9	10	
E-n22-k4(375)	混合遗传算法	60	50	1 317	1 449	1 567	1 165	1 139	1 033	1 683	1 669	1 473	899	
		100	1 000	395	375	382	382	387	390	383	375	386	386	
		200	1 000	378	377	375	375	375	375	382	375	376	375	
		100	10 000	375	375	375	382	375	375	375	375	375	375	
	本文算法	60	50	547	610	607	562	591	645	638	614	573	582	
		100	1 000	376	379	376	376	376	376	375	375	375	376	
		200	1 000	379	375	376	375	375	375	375	376	375	375	
		100	10 000	375	375	375	375	375	375	375	375	375	375	
	E-n33-k4(835)	混合遗传算法	100	1 000	931	882	940	912	916	905	911	889	928	867
			200	1 000	890	897	871	867	893	894	898	925	889	919
200			10 000	869	869	865	873	850	889	862	861	874	888	
500			10 000	845	846	862	846	837	862	837	837	842	852	
本文算法		100	1 000	864	873	868	848	853	873	863	877	868	870	
		200	1 000	853	856	859	878	849	862	872	851	865	880	
		200	10 000	844	839	839	842	846	839	847	842	839	838	
		500	10 000	838	839	842	839	839	839	839	839	842	837	

由表 3 可见,对于 21 个分仓库的实例 E-n22-k4 和 32 个分仓库的实例 E-n33-k4,随着种群规模和迭代次数的增加,两种算法求得解的质量也随之提高,但本文算法要明显优于文献[3]混合遗传算法的求解结果。对于 E-n22-k4,当种群规模为 100、迭代次数为 10 000 时,本文算法 10 次计算均能找到目前已知最优解,最优解线路详细调度为:第一辆车:中心仓库出发到达分仓库 9、7、5、2、1、6,最后返回中心仓库;第二辆车:中心仓库出发到达分仓库 13、11、4、3、8、10,最后返回中心仓库;第三辆车:中心仓库出发到达分仓库 17、20、18、15、12,最后返回中心仓库;第四辆车:中心仓库出发到达分仓库 16、19、21、14,最后返回中心仓库,线路图如图 2 所示。对于 E-n33-k4,当种群规模为 500、迭代次数为 10 000 时,本文算法和混合遗传算法均没有求得目前已知最优解,但都得到了非常接近已知最优解的一个近似解 837,其对应的线路详细调度为:第一辆车:中心仓库出发到达分仓库 30、14、31,最后返回中心仓库;第二辆车:中心仓库出发到达分仓库 3、5、6、10、18、19、21、20、22、23、24、25、17、13,最后返回中心仓库;第三辆车:中心仓库出发到达分仓库 29、28、16、27、26、15、1,最后返回中心仓库;第四辆车:中心仓库出发到达分仓库 2、12、11、32、8、9、7、4,最后返回中心仓库,线路图如图 3 所示。

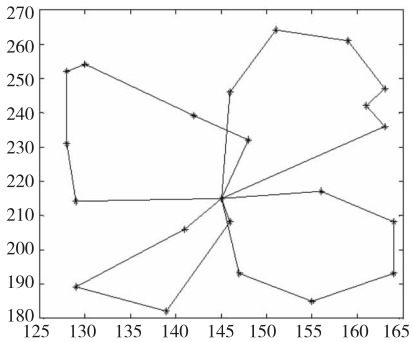


Figure 2 Vehicle routing of E-n22-k4

图 2 E-n22-k4 的车辆线路

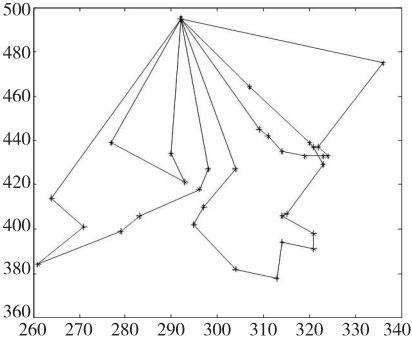


Figure 3 Vehicle routing of E-n33-k4

图 3 E-n33-k4 的车辆线路

5.3 实例 3

进一步验证本文算法对不同规模的车辆路径问题是否有效,对车辆路径问题数据库中其它一些不同规模的实例进行了仿真实验,并与解决该问题

较好的蚁群算法进行了比较,说明了本文算法对求解不同规模的车辆路径问题依然有效。限于篇幅,在这里我们只给出了部分比较结果,如表 4 所示。

Table 4 Results of the different scale vehicle routing problems
表 4 不同规模车辆路径问题的计算结果

实例名 (已知最优解)	蚁群算法	本文算法
A-n33-k5(661)	780	678
A-n45-k7(1146)	1 415	1 265
A-n55-k9(1073)	1 532	1 132
A-n65-k9(1174)	1 855	1 223
A-n80-k10(1763)	2 504	1 864
E-n101-k8(817)	1 413	859

从表 4 可以看出,当车辆路径问题的规模越来越大时,本文算法比蚁群算法有更大的优越性,且本文算法找到的解与目前已知最优解相差较小(以 E-n101-k8 为例,本文算法得到的解与已知最优解的误差仅为 5.14%),而这些已知最优解通常是几种启发式算法嵌套甚至借助了某些经典算法才求得的结果。

6 结束语

针对车辆路径问题,提出了一种食物源采用自然数编码的人工蜂群算法。算法利用邻域倒位操作生成候选食物源,采用罚函数处理不可行路径,仿真实验表明,本文算法可以有效求解车辆路径问题。同时,本文算法简单易行、方便操作,可以用来求解许多可以转化为车辆路径问题的组合优化问题。

参考文献:

[1] Jiang Da-li, Yang Xi-long, Du Wen, et al. A study on the genetic algorithm for vehicle routing problem[J]. Systems Engineering-Theory & Practice, 1999, 19(6):40-45. (in Chinese)

[2] Zhao Yan-wei, Wu Bin, Jiang Li, et al. Double populations genetic algorithm for vehicle routing problem[J]. Computer Integrated Manufacturing Systems, 2004, 10(3):303-306. (in Chinese)

[3] Jiang Chang-hua, Dai Shu-gui, Hu You-hua. Hybrid genetic algorithm for capacitated vehicle routing problem[J]. Computer Integrated Manufacturing Systems, 2007, 13(10):2047-2052. (in Chinese)

[4] Zhou Yan-cong, Sun Xiao-chen, Yu Wei-xiang. Research on optimization of logistics distribution route based on improved genetic algorithm [J]. Computer Engineering & Science, 2012,34(10):118-122. (in Chinese)

[5] Xiao Jian-mei, Li Jun-jun , Wang Xi-huai. Modified particle swarm optimization algorithm for vehicle routing problem [J]. Computer Integrated Manufacturing Systems, 2005, 11(4):577-581. (in Chinese)

[6] Yua B, Yanga Z Z, Yao B Z. An improved ant colony optimization for vehicle routing problem[J]. European Journal of Operational Research, 2009,196:171-176.

[7] Zhang X X, Tang L X. A new hybrid ant colony optimization algorithm for the vehicle routing problem[J]. Pattern Recognition Letters, 2009,30:848-855.

[8] Zhang Ze-bin, Hao Zhi-feng, Huang Han, et al. Ant colony optimization algorithm for capacitated vehicle routing problem based on multiple neighborhood descent searching[J]. Journal of Nanjing University (Natural Sciences), 2012, 48(1):91-98. (in Chinese)

[9] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Technical Report-TR06, Kayseri: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[10] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3):459-471.

[11] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization[J]. Foundations of Fuzzy Logic and Soft Computing, 2007, 4529:789-798.

[12] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm [J]. Applied Soft Computing, 2008, 8(1):687-697.

[13] Karaboga D. A new design method based on artificial bee colony algorithm for digital IIR filters[J]. Journal of the Franklin Institute, 2009, 346(4):328-348.

[14] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. Applied Soft Computing, 2009, 9(2):625-631.

[15] Hu Zhong-hua, Zhao Min. Research on robot path planning based on ABC algorithm [J]. Electric Welding Machine, 2009,39(4):93-96. (in Chinese)

[16] Hu Zhong-hua, Zhao Min. Simulation on traveling salesman problem(TSP) based on artificial bee colony algorithm[J]. Transactions of Beijing Institute of Technology, 2009, 29(11):978-982. (in Chinese)

[17] Sun Xiao-ya, Lin Yan. Improved artificial bee colony algorithm for assignment problem[J]. Microelectronics & Computer, 2012, 29(1):23-26. (in Chinese)

[18] Toth P, Vigo D. The vehicle routing problem[M]. PA:Society for Industrial and Applied Mathematics philadelphia, 2002.

附中文参考文献:

[1] 姜大立, 杨西龙, 杜文, 等. 车辆路径问题的遗传算法研究 [J]. 系统工程理论与实践, 1999, 19(6):40-45.

- [2] 赵燕伟, 吴斌, 蒋丽, 等. 车辆路径问题的双种群遗传算法求解方法[J]. 计算机集成制造系统, 2004, 10(3):303-306.
- [3] 姜昌华, 戴树贵, 胡幼华. 求解车辆路径问题的混合遗传算法[J]. 计算机集成制造系统, 2007, 13(10):2047-2052.
- [4] 周艳聪, 孙晓晨, 余伟翔. 基于改进遗传算法的物流配送路径优化研究[J]. 计算机工程与科学, 2012, 34(10):118-122.
- [5] 肖健梅, 李建军, 王锡淮. 求解车辆路径问题的改进微粒群优化算法[J]. 计算机集成制造系统, 2005, 11(4):577-581.
- [8] 张泽彬, 郝志峰, 黄翰, 等. 求解车辆路径问题的多邻域下降搜索蚁群优化算法[J]. 南京大学学报(自然科学版), 2012, 48(1):91-98.
- [15] 胡中华, 赵敏. 基于人工蜂群算法的机器人路径规划[J]. 电焊机, 2009, 39(4):93-96.
- [16] 胡中华, 赵敏. 基于人工蜂群算法的 TSP 仿真[J]. 北京理工大学学报, 2009, 29(11):978-982.
- [17] 孙晓雅, 林焰. 改进的人工蜂群算法求解任务指派问题[J]. 微电子学与计算机, 2012, 29(1):23-26.

作者简介:



王志刚(1978-),男,山东高密人,硕士,讲师,研究方向为组合优化与算法研究。**E-mail:** wzg19.scut@163.com

WANG Zhi-gang, born in 1978, MS, lecturer, his research interests include combination optimization, and algorithm research.



夏慧明(1981-),男,江苏泰州人,硕士,讲师,CCF 会员(E200014698M),研究方向为组合优化与算法研究。**E-mail:** Huimingxia1981@163.com

XIA Hui-ming, born in 1981, MS, lecturer, CCF member(E200014698M), his research interests include combination optimization, and algorithm research.

2014 年全国理论计算机科学学术年会征文通知

由中国计算机学会主办、山东大学计算机科学与技术学院承办的“2014 年全国理论计算机科学学术年会”将于 2014 年 10 月 16 至 17 日在山东省济南市召开。有关征文要求通知如下:

为进一步提高学术水平,本次年会将采取以下措施:①除征集尚未在其他刊物或学术会议上正式发表过的论文外,还将征文扩大到 2013、2014 年中国大陆学者在国际一流学术刊物上已发表的论文(注明出处),对后者经审稿录用后,邀请作者出席年会并作报告(但论文不再收入论文集);②除特邀报告外,还将安排专委会委员作大会报告,并组织专题讨论会。

征文范围

- (1)程序理论(程序逻辑、程序正确性验证、形式化开发方法、高可信软件等)
- (2)计算理论(算法设计与分析、复杂性理论、可计算性理论、量子计算、生物信息学、社会网络等)
- (3)语言理论(形式语言理论、自动机理论、形式语义学、计算语言学等)
- (4)人工智能(知识工程、机器学习、模式识别、机器人、数据挖掘、进化计算、脑机接口等)
- (5)逻辑基础(数理逻辑、多值逻辑、模糊逻辑、模态逻辑、直觉主义逻辑、组合逻辑等)
- (6)数据理论(演绎数据库、关系数据库、面向对象数据库、分布式数据库、空间数据库、大数据理论等)
- (7)计算机数学(符号计算、数学定理证明、计算几何、组合数学等)
- (8)并行算法(分布式并行算法、大规模并行算法、网格算法、多核并行算法、GPU 上并行算法等)

本次会议可接受中文或英文论文。会议录用的英文论文将在《Journal of Computational Information Systems》正刊上发表,录用的中文论文将在《计算机科学》、《计算机工程与科学》和《计算机与数字工程》正刊上发表。会议将推荐优秀论文(第一作者为学生)到《计算机研究与发展》的正刊上发表。

论文稿件的排版说明请参见会议网站:<http://nctcs2014.sdu.edu.cn>。

论文投稿截止日期:2014 年 5 月 20 日;论文录用通知日期:2014 年 7 月 1 日。

论文投稿需网上提交,投稿网站为 <https://www.easychair.org/conferences/?conf=nctcs2014>,请按照要求提交电子版 WORD 文件或者 PDF 文件。大会提供的 Email 信箱为:nctcs2014@126.com。

会议动态和重要信息请关注会议网站:<http://nctcs2014.sdu.edu.cn>。

联系人和联系电话及 Email 信箱:

朱大铭:nctcs2014@126.com;

联系电话:0531-88390006;

联系地址:山东省济南市舜华路 1500 号,山东大学计算机科学与技术学院,邮编:250101。

中国计算机学会理论计算机科学专业委员会
山东大学计算机科学与技术学院

2014 年 2 月 18 日