

DOI:10.3724/SP.J.1008.2009.00074

基于单亲遗传算法求解卫勤保障最短回路问题

蒋兴波¹, 许开云², 刘晓荣^{1*}

- 1. 第二军医大学卫生勤务学系卫生勤务学教研室, 上海 200433
- 2. 第二军医大学长海医院急诊科, 上海 200433

[摘要] **目的** 提出一种用于求解卫勤保障最短回路问题的有效算法。**方法** 采用仅使用变异算子和选择算子来繁殖后代的单亲遗传算法(PGA)。该算法基于K-随机近邻算法,同时结合了随机两点位置互换、相邻两点位置互换、环形部分逆转、随机插入等4种变异算子,在选择操作过程中结合贪心策略,进一步提高PGA算法的“爬山”能力。**结果** 通过对CTSP31以及多组TSP标准数据集的测试结果表明,该算法相对于文献中已有算法更加有效。**结论** 该算法可作为今后开发计算机辅助软件的基础,为提高卫勤指挥的质量和速度而提供辅助优化决策。

[关键词] 卫勤保障最短路径问题;旅行商问题;单亲遗传算法;K-随机近邻算法

[中图分类号] R 821.4 **[文献标志码]** A **[文章编号]** 0258-879X(2010)01-0074-06

Solving shortest circuit problem of military health support based on partheno-genetic algorithm

JIANG Xing-bo¹, XU Kai-yun², LIU Xiao-rong^{1*}

- 1. Department of Military Health Service, Faculty of Health Services, Second Military Medical University, Shanghai 200433, China
- 2. Department of Emergency Medicine, Changhai Hospital, Second Military Medical University, Shanghai 200433, China

[Abstract] **Objective** To propose an effective algorithm for solving the shortest circuit problem of military health support. **Methods** Partheno-genetic algorithm (PGA), which only used mutation operation and selection operation, was adopted in the present study. The algorithm was based on K-random-nearer-neighbor algorithm combined with two-random-point exchange, two-neighbor-point exchange, circular-based part inversion and random insertion mutation operations. Furthermore, greedy strategy was applied in selection to improve the hill-climbing capability of PGA. **Results** The simulation results of CTSP31 and standard dataset from TSP library indicated that the PGA was more effective than existing algorithms from the literature. **Conclusion** PGA can serve as a basis for further development of a computer-assisted program, and it provides optimized decision-making scheme for improving the quality and speed of military medical service disposition.

[Key words] the shortest circuit problem of health support; traveling salesman problem; partheno genetic algorithm; K-random-nearer-neighbor algorithm

[Acad J Sec Mil Med Univ, 2010, 31(1):74-79]

按照未来信息化战争的要求,我军后勤的发展方向将是配送式后勤,对于卫勤保障机构而言,未来的发展方向也将是配送式的。因此卫勤物资保障机构可以理解为战时条件下的卫勤物资配送中心。同时,为了减少战时伤病员的阵亡率和伤死率,提高伤病员的归队率,大量装备轻便、技术精湛、机动灵活的医疗小分队将出现在伤病员抢救的第一线,美军称之为前沿手术队。未来战争中除了炸伤、烧伤较多外,多部位伤也将大量出现,这不但给伤员的救治带来了难度,同时将消耗大量的药品器材。因此,如

何将卫勤物资配送中心的各类卫勤资源快速空投或汽运至各个医疗点,对医疗小分队及时高效完成救死扶伤任务起到至关重要的作用。而要实现资源的快送,路径的选择则是首要考虑因素,这就是卫勤保障最短回路问题(the shortest circuit problem of health support, SCPHS)。

SCPHS属于典型的旅行商问题(traveling salesman problem, TSP),它是完全多项式非确定性(nondeterministic polynomial time complete, NPC)问题,其算法复杂度随着问题规模的增加呈现指数

[收稿日期] 2009-07-01 **[接受日期]** 2009-11-09

[基金项目] 全军“十一五”科技攻关课题(08G078)。Supported by Key Research Project of “11th Five-year” of PLA(08G078)。

[作者简介] 蒋兴波, 硕士, 讲师。E-mail: jiangxb. smmu@gmail.com

* 通讯作者(Corresponding author)。Tel: 021-81871423, E-mail: lxr1966@yahoo.com.cn

增长。针对 TSP 问题, 相关学者进行了大量的研究, 采用了各种优化算法, 如模拟退火算法 SA (simulated annealing algorithm)^[1-3]、蚁群算法 ACO (ant colony algorithm)^[3-6]、粒子群算法 PSO (particle swarm optimization algorithm)^[7]、Hopfield 神经网络方法^[8-9]以及遗传算法 GA^[3,7,10-15]等对其进行求解。而遗传算法 GA (genetic algorithm) 由于具有良好的全局搜索能力, 已成为目前解决各种优化问题的有效工具之一。

本文提出了一种只使用变异算子和选择算子来繁殖后代的单亲遗传算法 (partheno genetic algorithm, PGA), 该算法采用了整数编码方式, 用 K-随机近邻算法产生初始群体, 在随机两点位置互换、相邻两点位置互换变异算子的基础上, 加入了环形部分逆转、随机插入变异算子, 同时在排序选择操作上, 结合了最优保存策略 (elitist model) 来选择生成下一代个体。

1 数学模型的建立

SCPHS 是一个组合优化问题, 可以描述为: 给定 $n-1$ 个需求点 (医疗小分队) 和一个供应点 (配送中心), 寻找一条闭合路径, 使得每个点 (包括需求点和供应点) 刚好经过一次且总的行驶距离最短。其数据模型描述如下:

有 n 个节点:

$$P = \{p_1, p_2, \dots, p_n\}$$

其中任意点 P_i 的坐标为 $(P_i \cdot x, P_i \cdot y)$, 任意两点 P_i, P_j 之间的距离为

$$d(p_i, p_j) = \sqrt{(p_i \cdot x - p_j \cdot x)^2 + (p_i \cdot y - p_j \cdot y)^2}$$

即寻找一条闭合路径 $I = \{\pi(1), \pi(2), \dots, \pi(n)\}$, 使得下列目标函数最小:

$$D(I) = \sum_{i=1}^{n-1} d(p_{\pi(i)}, p_{\pi(i+1)}) + d(p_{\pi(n)}, p_{\pi(1)}) \quad (1)$$

即求 $\min\{D(I)\}$ 。

最短回路问题与最短路径问题不同, 最短路径问题是求网络中两点之间的距离最短的路径, 该问题能够在多项式时间内得到全局最优解, 针对该问题的算法较多而且已经很成熟, 如非常经典的 Dijkstra 算法^[16]等; 而最短回路问题则是寻找一条经过所有点的回路问题, 该问题的算法复杂度为指数, 即在多项式时间内无法找到最优解。通常情况下, 当问题规模较大 (即经过点较多) 时, 往往很难获得全局最优解。

2 算法的实现

PGA 是一种适合求解组合问题的新型遗传算

法, 它取消了传统遗传算法中的交叉算子, 而只在同一条染色体上采用基因变异等方法进行基因重组, 因此相对于传统遗传算法, 该算法具有操作简单、计算效率高、收敛速度快等优点。

2.1 编码及适应度函数 SCPHS 采用整数编码 (encoding), 即由 1 个供应点和 $n-1$ 个需求点共 n 个节点分别用整数 $1, 2, \dots, n$ 编号, 一条完整的闭合回路对应一个染色体编码, $I_j = \{\pi(1), \pi(2), \dots, \pi(n)\}$, 其中 $\pi(i)$ 表示节点编号 $0 \leq \pi(i) \leq n$; I_j 表示第 j 个个体。

例如: 假设现有 9 个需求点和 1 个供应点, 则编码后染色体的长度为 10。如个体 I_j 的染色体编码为 (1 9 7 2 4 5 3 6 8 10), 表示从第 1 号节点开始, 依次经过点 9-7-2-4-5-3-6-8-10, 最后返回到 1 号节点, 从而形成一条闭合回路。

在 PGA 中, 适应度函数 (fitness function) 用于对个体进行评价, 它是个体进化的依据。个体的适应度越高, 其被选择的概率就越大。本文的适应度函数由环路路径长度的倒数表示, 即

$$f(I_j) = 1/D(I_j) \quad (2)$$

从公式 (2) 中可以看出, 当环路路径越长, 其个体对应的适应度就越低, 其被选择的概率就越小; 当环路路径越短, 其个体对应的适应度就越高, 其被选择的概率就越大。

2.2 初始群体的产生 一般而言, 对 NP 问题很难直接构造出一个最优解或满意解, 所以只能通过搜索方法在整个解空间寻找最优解或满意解。当问题规模较大时, 这种盲目搜索或者遍历就变得十分困难, 甚至根本不可行。因此相关的研究学者常利用问题解本身的某些结构特征来指导对问题解的搜索。

SCPHS 存在这样的特征, 即在最佳路径的选取中必然包括而且在很大程度上包括相邻节点间最短的路径。根据这一特征, 节点 P_i 的下一个节点 P_{i+1} 应主要从其最近领域中选取。同时为了排除最优路径中个别节点的下一个节点不在其选取的最近领域内, 本研究提出了一种新的产生单个个体算法, 即 K-随机近邻算法 (K-random-nearer-neighbor algorithm), 其流程如下:

Step1: $i=1$;

Step2: 从集合 $C = \{1, 2, \dots, n\}$ 随机选取一个元素作为染色体的第一个基因 $\pi(i)$, 并删除 C 中该元素;

Step3: 判断 C 是否为空, 如为空, 转 Step6; 否则继续;

Step4: 在 C 中随机选择 K 个元素, 并计算这 K

个元素对应节点到 $P_{\pi(i)}$ 的距离,将离 $P_{\pi(i)}$ 最近点对应元素作为染色体的下一个基因 $\pi(i+1)$,并将该元素从集合 C 中删除;

Step5: $i=i+1$,转 Step3;

Step6:产生一个新的个体 I ,算法结束。

算法中,K的取值也较为关键,当K较大时,比较次数较多,算法的效率降低;当K较小时,其整体运行效果较差。通过多组数据多次实验,K取5时能取得相对较好的结果。

PGA的初始群体可由K-随机近邻算法重复运行多次产生。

2.3 变异 本算法中采用了随机两点位置互换、相邻两点位置互换、环形部分逆转、随机插入^[17]等4种变异算子。在变异点的选择过程中结合贪心策略,来进一步提高PGA算法的“爬山”能力。

2.3.1 随机两点位置互换 随机两点位置互换(two-random-point exchange)变异的实现流程如下:

Step1:随机生成两个变异点 $i, j \in [1, n] (i \neq j)$;

Step2:计算变异前和变异后目标函数值是否有改进,如有改进,即 $d(p_{\pi(i-1)}, p_{\pi(i)}) + d(p_{\pi(i)}, p_{\pi(i+1)}) + d(p_{\pi(j-1)}, p_{\pi(j)}) + d(p_{\pi(j)}, p_{\pi(j+1)}) > d(p_{\pi(i-1)}, p_{\pi(j)}) + d(p_{\pi(j)}, p_{\pi(i+1)}) + d(p_{\pi(j-1)}, p_{\pi(i)}) + d(p_{\pi(i)}, p_{\pi(j+1)})$,则将两个基因互换;

Step3:判断是否循环 n 次,如是,则退出;否则转 Step1。

例如: $I_j = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i), \pi(i+1), \dots, \pi(j-1), \pi(j), \pi(j+1), \dots, \pi(n)\}$,则变异后产生的新个体为 $I_j' = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(j), \pi(i+1), \dots, \pi(j-1), \pi(i), \pi(j+1), \dots, \pi(n)\}$ 。

2.3.2 相邻两点位置互换 相邻两点位置互换(two-neighbor-point exchange)变异的实现流程如下:

Step1:随机生成一个变异 $i \in [1, n]$;

Step2:计算变异前和变异后目标函数值是否有改进,如有改进,即 $d(p_{\pi(i-1)}, p_{\pi(i)}) + d(p_{\pi(i+1)}, p_{\pi(i+2)}) > d(p_{\pi(i-1)}, p_{\pi(i+1)}) + d(p_{\pi(i)}, p_{\pi(i+2)})$,则将两个基因互换;

Step3:判断是否循环 n 次,如是,则退出;否则转 Step1。

例如: $I_j = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i), \pi(i+1), \pi(i+2), \dots, \pi(n)\}$,则变异后产生的新个体为 $I_j' = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i+1), \pi(i), \pi(i+2), \dots, \pi(n)\}$ 。

2.3.3 环形部分逆转 环形部分逆转(circular-based part inversion)变异的实现流程如下:

Step1:随机生成变异点 $i \in [0, n]$;

Step2:随机生成变异长度 $l \in [0, n/2]$;

Step3:计算变异前和变异后目标函数值是否有改进,如有改进,即 $d(p_{\pi(i-1)}, p_{\pi(i)}) + d(p_{\pi((i-1+l) \bmod(n))}, p_{\pi((i+l) \bmod(n))}) > d(p_{\pi(i-1)}, p_{\pi((i+l) \bmod(n))}) + d(p_{\pi((i-1+l) \bmod(n))}, p_{\pi(i)})$,则将 i 后的 l 位基因逆转;

Step4:判断是否循环 n 次,如是,则退出;否则转 Step1。

例如:父代个体 $I_i = (6\ 2\ 1\ 8\ 4\ 5\ 9\ 3\ 7)$,起始变异点 $i=6$,变异长度 $l=5$,则变异后产生的新个体 $I_i' = (3\ 9\ 1\ 8\ 4\ 5\ 2\ 6\ 7)$ 。

2.3.4 随机插入 随机插入(random insertion)变异的实现流程如下:

Step1:随机生成两个变异点 $i, j \in [1, n] (i \neq j)$;

Step2:计算变异前和变异后目标函数值是否有改进,如有改进,即 $d(p_{\pi(i-1)}, p_{\pi(i)}) + d(p_{\pi(i)}, p_{\pi(i+1)}) + d(p_{\pi(j-1)}, p_{\pi(j)}) > d(p_{\pi(i-1)}, p_{\pi(i+1)}) + d(p_{\pi(j-1)}, p_{\pi(i)}) + d(p_{\pi(i)}, p_{\pi(j+1)})$,则将两个基因互换;

Step3:判断是否循环 n 次,如是,则退出;否则转 Step1。

例如: $I_j = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i), \pi(i+1), \dots, \pi(j-1), \pi(j), \dots, \pi(n)\}$,则变异后产生的新个体为 $I_j' = \{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i+1), \dots, \pi(j-1), \pi(i), \pi(j), \dots, \pi(n)\}$ 。

2.4 选择 本文采用排序选择方法(rank-based select methods)^[18],该方法的主要优点是个体适应度是否取正值或负值以及个体适应度之间的数值差异程度无关,而只与个体适应度值之间的大小关系相关。其步骤如下:

Step1:对群体中所有个体按适应度从大到小排序;

Step2:按下列公式计算每个个体的选择概率:

$$P_s^{(i)} = a^i \tag{3}$$

其中 $P_s^{(i)}$ 表示排列在第 i 个位置个体的选择概率, a 为调节参数。

Step3:根据 Step2 计算得出的个体概率值作为其能够被遗传到下一代的概率,基于该值用比例选择(赌盘选择)的方法产生下一代群体。

为了避免上一代产生的最佳个体丢失,本文还混合使用了最优保存策略,即将当前代中适应度最好的个体保留到下一代群体,从而在一定程度上保证遗传算法的全局收敛性,而且该精英个体可能会产生适应度更高的新个体。

2.5 终止条件 PGA在实际应用中是不允许无限地循环下去,而且通常情况下问题的最优解事先

未必知道,因此需要有一定的条件来终止算法的进程。

常见的终止条件包括事先设定一个最大的进化代数、或者判断最佳解是否满足要求、或者判断最佳解是否连续若干步没有明显变化等。本文中我们采用固定的循环代数和是否获得最优解相结合的方式。

2.6 算法的总体流程 本文将环形交叉算子和环形变异算子应用于遗传算法中,并在变异操作中结合贪心策略来解决 SCPHS,其算法流程如下:

Step1:个体染色体编码;

Step2: $k=0$;根据 K-随机近邻算法产生 m 个个体组成初始群体 $pop(k)=\{I_1, I_2, \dots, I_m\}$;

Step3:对单个个体计算其总的距离 $D(I_i)$;记录个体适应度 $f(I_i)(f(I_i)=1/D(I_i))$;

Step4:判断是否满足停止条件。若是,则停止计算,输出最佳结果;否则继续;

Step5:利用排序选择操作选择 m 个个体形成新的群体 $selpop(k+1)$;

Step6:根据变异概率 P_m 对所有个体依次进行随机两点位置互换、相邻两点位置互换、环形部分逆转、随机插入等 4 类变异操作,产生群体 $mutpop(k+1)$;

Step7: $pop(k+1)=mutpop(k+1)$; $k=k+1$;返回 Step3。

3 算法的验证

为了验证算法的可行性和有效性,本文进行了两组测试,一组为国内常用测试集,一组为 TSPLIB^[19] 标准测试集。实验平台为:ThinkPad R400 双核处理器 CPU 2.4G,2G RAM,Windows Vista。算法用 VC++ 编程实现。

PGA 的运行参数设置为:变异概率 $P_m=1.0$,群体规模 $PopSize=$ 节点数 n ,选择概率调节参数 $a=0.955$,进化代数 $k=1000$ 。

3.1 实验 1 测试集来自文献[15],其对应的是中国旅行商问题(chinese traveling salesman problem, CTSP),即在中国 31 个主要城市(包括省会城市、自治区首府和直辖市)之间进行旅游,找出一条最短的经过每个城市仅一次的且回到起点的路径。由于该问题给出的 31 个城市具有实际意义且分布具有随意性,因此常被用作检验求解 TSP 的各种算法优劣的经典测试集。

对该测试集用 PGA 重复测试 10 次,实验对比

结果见表 1,其中 D_{\min} 、 D_{\max} 、 D_{avg} 分别表示算法多次测试中求得的最好解、最差解以及平均解, T_{avg} 表示平均运行时间。

表 1 10 种算法对 CTSP31 的实验结果对比

Tab 1 Results comparison of 10 algorithms and CTSP31

Algorithm	D_{\min}	D_{\max}	D_{avg}	$T_{\text{avg}} t/s$
GA ^[3]	16 369	—	18 216	20.44
ACO ^[3]	15 602	—	15 648	69.82
SA ^[3]	15 567	—	16 981	12.38
SA ^[6]	15 398	18 247	16 902	—
GA ^[6]	15 387	19 298	16 920	—
HPSO ^[6]	15 383	16 649	15 733	—
SLPSO ^[6]	15 381	15 398	15 381.6	—
GA ^[12]	15 378	15 795	15 480	32
PMA ^[14]	15 377	—	—	—
PGA	15 377	15 377	15 377	0.60

GA: Genetic algorithm; ACO: Ant colony algorithm; SA: Simulated annealing algorithm; HPSO: Hybrid particle swarm optimization algorithm; SLPSO: Self-learning particle swarm optimization algorithm; PMA: Population migration algorithm; PGA: Partheno genetic algorithm

对 CTSP31,目前文献给出的最好结果是 15 377^[14]。从表 1 可以看出,PGA 算法在 10 次测试中皆取得了该最好解,除廉侃超等^[14]在多次测试中曾获得最好解之外,其他算法求得的最好结果皆大于 15 377。实验结果证明,本文提出的算法是有效的。

3.2 实验 2 采用 TSPLIB^[19] 中的部分标准数据集作为测试集,其规模 n 从 17~200 不等。对这些测试集,同样用本文的 PGA 算法重复运行 10 次,并记录不同实例(instance)获得的到最优解的最小偏差 $E_{\min}(\%)$ [(某实例多次测试获得的最好解 D_{\min} —该实例 TSPLIB 中给出的全局最优解 $D_{\text{opt}})/D_{\text{opt}} \times 100\%$]、平均偏差 $E_{\text{avg}}(\%)$ [(某实例多次测试获得的平均解 D_{avg} —该实例 TSPLIB 中给出的全局最优解 $D_{\text{opt}})/D_{\text{opt}} \times 100\%$],并将实验结果分别与文献[7]和文献[13]的结果进行比较,结果见表 2、表 3。

文献[13]中,Yang 等给出了一种广义染色体遗传算法(generalized chromosome genetic algorithm, GCGA),该算法分为不带局部搜索(GCGA without local search, GCGA0)和带局部搜索(GCGA with local search, GCGA1)两类,并将这两类算法和经典遗传算法(classical genetic algorithm, CGA)进行了实验对比。

表 2 GA、ACO 与 PGA 的实验对比

Tab 2 Result comparison of GA, ACO and PGA

Instance	n	D _{opt}	PGA		GA ^[7]		ACO ^[7]	
			E _{min} (%)	E _{avg} (%)	E _{min} (%)	E _{avg} (%)	E _{min} (%)	E _{avg} (%)
Gr17	17	2 085	0	0	0	0.048	3.549	3.597
Gr21	21	2 707	0	0	0	0.517	8.903	9.937
Gr24	24	1 272	0	0	0	1.415	2.752	9.198
Fri26	26	937	0	0	0	3.095	0	4.376
Bays29	29	2 020	0	0	0	1.436	0	1.634
Oliver30	30	423.74	0	0	0	0.911	0	6.837
Swiss42	42	1 273	0	0	9.269	12.726	0	4.085
Eil51	51	426	0	0.070	16.197	20.188	0	3.286
St70	70	675	0	0	46.074	59.704	0	4.000
Average error			0	0.008	7.949	11.116	1.689	5.217

PGA:Partheno genetic algorithm; GA:Genetic algorithm; ACO:Ant colony algorithm

表 3 GCGA0、GCGA1、CGA 与 PGA 的实验对比

Tab 3 Result comparison of GCGA0, GCGA1, CGA and PGA

Instance	n	D _{opt} (%)	PGA		GCGA0 ^[13]		GCGA1 ^[13]		CGA ^[13]	
			E _{min} (%)	E _{avg} (%)	E _{min} (%)	E _{avg} (%)	E _{min} (%)	E _{avg} (%)	E _{min} (%)	E _{avg} (%)
Eil51	51	426	0	0.07	1.17	2.58	0.24	0.94	0.94	2.11
St70	70	675	0	0	1.48	2.07	0	0.44	1.19	3.41
Eil76	76	538	0	0.11	1.86	2.42	2.23	2.42	2.60	5.76
Pr76	76	108 159	0	0	1.45	1.84	0.14	0.72	0.90	1.90
Rat99	99	1 211	0	0.01	2.15	4.71	0.99	2.23	4.29	6.85
KroA100	100	21 282	0	0	1.22	2.28	0.05	1.23	1.28	3.54
KroB100	100	22 141	0	0.17	2.93	4.10	0.24	1.81	0.17	2.70
KroC100	100	20 749	0	0	0.68	3.45	0.32	1.33	2.02	3.79
KroD100	100	21 294	0	0.04	1.06	4.58	1.22	2.42	2.23	4.14
KroE100	100	22 068	0	0.20	1.60	3.42	0.24	1.41	2.97	4.11
Rd100	100	7 910	0	0.04	1.30	2.60	0.96	1.53	3.70	4.91
Eil101	101	629	0	0.70	3.82	5.09	1.59	2.70	5.72	6.68
Lin105	105	14 379	0	0	0.87	1.68	0.56	1.15	0.19	1.98
Pr107	107	44 303	0	0.15	0.50	2.18	1.15	1.37	1.49	2.93
Pr124	124	59 030	0	0	0.08	1.06	0	0.19	0.50	1.74
Bier127	127	118 282	0	0.72	1.89	2.83	0.91	1.80	2.28	3.22
Pr136	136	96 772	0	0.54	4.13	5.80	0.47	2.82	2.81	3.77
Pr144	144	58 537	0	0	0.03	0.74	0	0.04	0.17	0.78
KroA150	150	26 524	0	0.10	1.66	5.06	1.40	2.92	4.21	5.55
KroB150	150	26 130	0.04	0.38	2.97	4.70	1.63	2.11	2.17	3.51
Pr152	152	73 682	0	0.11	1.32	2.51	0.19	1.22	1.70	2.25
U159	159	42 080	0	0.30	5.24	7.02	1.74	2.28	5.13	6.26
Rat195	195	2 323	0.43	1.35	6.54	7.40	3.14	4.18	5.34	7.40
D198	198	15 780	0.02	0.21	1.81	3.26	1.01	1.93	2.08	2.75
KroA200	200	29 368	0.05	0.43	2.80	4.56	1.33	1.85	2.72	4.52
KroB200	200	29 437	0.01	0.42	5.27	6.77	2.09	4.04	5.03	6.21
Average error			0.02	0.23	2.15	3.64	0.92	1.81	2.46	3.95

GCGA:Generalized chromosome genetic algorithm; GCGA0;GCGA without local search;GCGA1;GCGA with local search;CGA:Classical genetic algorithm; PGA:Partheno genetic algorithm

从表 2 中可以看出,对 9 组实例的测试中,PGA 获得的 E_{min} 全部为 0,表示每组实例的 10 次实验中皆获得了最优解 D_{opt},而 GA 和 ACO 算法则只有 6

次获得对应最优解;同时 PGA 获得的 E_{avg} 中有 8 组实例为 0,表示该 8 组实例 10 次测试皆得到最优解,而 GA 和 ACO 算法的 E_{avg} 却全部大于 0。从总体的

平均偏差(average errors)可以看出,PGA 获得的结果解远远小于 GA 和 ACO 算法获得的结果。表 3 中,PGA 在对 26 组实验的测试中, E_{\min} 为 0 的有 21 组,而其他 3 种算法 GCGA0、GCGA1、CGA,除 GCGA1 有 3 组实例的 E_{\min} 为 0 外,其余结果皆大于 0;同样,PGA 有 7 组实例的 E_{avg} 为 0,而其他 3 种算法的 E_{avg} 全部大于 0。从平均偏差上看,PGA 的 E_{\min} 和 E_{avg} 也远远小于其他对应 3 种算法获得的结果。

实验结果证明,本文采用的 PGA 相对于其他算法,其结果是有效的。

4 讨论

对 SCPHS,目前相关研究还相对较少。对这类 NP 问题,往往只能通过启发式算法来获得最好解或者满意解。

本文中,我们采用了 PGA,在初始群体中提出了 K-随机近邻算法,并在变异中采用了随机两点位置互换、相邻两点位置互换、环形部分逆转、随机插入等 4 种变异算子。两组实验结果表明,该 PGA 算法是行之有效的。该算法可作为未来各类卫勤组织指挥软件的基础,为提高平时卫勤指挥效率提供辅助决策功能。

在测试过程中发现,当问题规模较大时(>200 个节点),算法的寻优能力特别是搜寻全局最优的能力还有待提高,而且算法的运行时间还相对较长。因此,如何进一步提高算法的效果和效率,仍将是下一步的研究重点。

[参考文献]

- [1] 高尚. 求解旅行商问题的模拟退火算法[J]. 华东船舶工业学院学报,2003,17:13-16.
- [2] 胡大伟,朱志强,胡勇. 车辆路径问题的模拟退火算法[J]. 中国公路学报,2006,19:123-126.

- [3] 李敏,吴浪,张开碧. 求解旅行商问题的几种算法的比较研究[J]. 重庆邮电大学学报(自然科学版),2008,20:624-626,630.
- [4] 吴斌,史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法[J]. 计算机学报,2001,24:1328-1333.
- [5] 吴义虎,李宁,杨秋实. 一种改进的蚁群算法及其在 TSP 中的应用[J]. 长沙交通学院学报,2007,23:32-35.
- [6] 蔡荣英,李丽珊,林晓宇,钟一文. 求解旅行商问题的自学习粒子群优化算法[J]. 计算机工程与设计,2007,28:261-263,266.
- [7] 蔡光跃,董恩清. 遗传算法和蚁群算法在求解 TSP 问题上的对比分析[J]. 计算机工程与应用,2007,43:96-98.
- [8] 陈萍,郭金锋. 对 Hopfield 神经网络求解 TSP 的研究[J]. 北京邮电大学学报,1999,22:58-61.
- [9] 程明,刘琴. 神经网络 TSP 问题仿真分析[J]. 郑州大学学报:理学版,2004,36:45-48.
- [10] 魏英姿,赵明扬,黄雪梅,胡玉兰. 求解 TSP 问题的贪心遗传算法[J]. 计算机工程,2004,30:19-20,34.
- [11] 冯春松,王军宇,周松盛,彭斯俊,王攀. TSP 问题的一种改进遗传算法[J]. 武汉理工大学学报,2006,28:116-118,130.
- [12] 柴世红. 两段式遗传算法求解 CTSP[J]. 大众科技,2008,4:17-19.
- [13] Yang J H, Wu C G, Lee H P, Liang Y C. Solving traveling salesman problems using generalized chromosome genetic algorithm[J]. Progr Nat Sci,2008,18:887-892.
- [14] 廉侃超,古明家,李永胜,宣士斌. 求解 TSP 问题的嵌入交叉算子的人口迁移算法[J]. 计算机工程,2008,34:224-226.
- [15] 李明海,刑桂华. 用 MATLAB 实现中国旅行商问题的求解[J]. 微计算机应用,2004,25:218-222.
- [16] 屈婉玲,耿素云,王捍贫. 离散数学教程[M]. 北京:北京大学出版社,2004:201-203.
- [17] 赵政,薛桂香,宋建材,孟和. 基于动态遗传算法的网格计算任务调度[J]. 计算机工程,2008,34:191-193.
- [18] Baker J E. Adaptive selection methods for genetic algorithm [C]//Proceedings of an international conference on genetic algorithms and their application. New Jersey:Lawrence Erlbaum Associate,1985:101-111.
- [19] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html>. [2009-03-12].

[本文编辑] 尹茶