

基于改进PSO算法的机动通信保障任务分配方法

滑楠, 赵延龙[†], 于振华

(空军工程大学 信息与导航学院, 西安 710077)

摘要: 针对机动通信保障问题建立任务分配模型, 结合梯度下降法提出一种基于改进粒子群算法(TSPSO)的任务分配模型求解方法. 在TSPSO算法中增加判断极值陷阱、粒子二次搜索、设定禁忌区域、粒子淘汰与生成4个部分, 并将TSPSO算法与其他4种改进PSO算法应用于四种典型测试函数的优化. 结果表明, TSPSO算法收敛精度更高、收敛速度更快. 在基于TSPSO算法的任务分配模型求解方法中, 基于各机动通信保障单元到不同通信地点分配概率的思想对粒子群进行编码和解码, 提高模型求解效率. 仿真结果表明, TSPSO算法能够快速寻找到机动通信保障任务最优分配方案.

关键词: 机动通信保障; 任务分配; 粒子群算法; 二次搜索; 分配概率

中图分类号: TP301.6

文献标志码: A

Method of task allocation of tactical communication support based on improved particle swarm optimization algorithm

HUA Nan, ZHAO Yan-long[†], YU Zhen-hua

(College of Information and Navigation, Air Force Engineering University, Xi'an 710077, China)

Abstract: For solving the problem of tactical communication support, a task allocation model is established, and a twice search particle swarm optimization(TSPSO) algorithm based on gradient descent is proposed. There are four parts added into the TSPSO algorithm, including determination of the extremum trap, particle twices search, set forbidden area, particle elimination and generated. The TSPSO algorithm and other four improved algorithms are applied to the optimization problem of four typical test functions. The results show that, the convergence accuracy of the TSPSO algorithm is higher, the convergence speed is faster. In the solution of the task allocation model, the probability distribution of the support units to communication places is used to encode and decode the particle swarm. The simulation results show that the proposed algorithm can quickly find the optimal allocation scheme of tactical communication support tasks.

Keywords: tactical communication support; task allocation; PSO; twice search; allocation probability

0 引言

机动通信保障任务分配是安排有限的机动通信保障单元有效地完成对若干个通信地点的保障任务, 优化目标是总的效费比最优. 解决此类问题需要综合考虑实际环境中多种制约因素, 属于NP难问题. 传统的解决方法有整数规划法^[1-2]、随机搜索法^[3]等, 然而当模型中决策变量和约束条件较多时, 这些方法存在时间复杂度过高的问题, 不能满足实际保障环境中实时任务分配的需求. 近年来随着群智能优化技术的发展, 遗传算法^[4-6](Genetic algorithm, GA)、蚁群算法^[7-8](Ant colony optimization, ACO)、人工蜂群算法^[9](Artificial bee colony, ABC)等被应用于多种

领域的任务分配中, 取得了较好的效果.

粒子群算法^[10](Particle swarm optimization, PSO)是1995年由Eberhart和Kennedy提出的一种群体智能优化算法. 该算法以其实现简便、精度高、收敛快等特点在解决科学研究和理论计算等方面展示了其优越性^[11-12].

本文依据实际保障环境制约因素建立机动通信保障任务分配模型, 通过对PSO中粒子的搜索行为进行研究, 结合梯度下降法的快速收敛特性, 提出一种基于二次搜索的改进粒子群优化算法(Twice search particle swarm optimization, TSPSO), 并应用于任务分配模型的求解. 最后通过具体实例对任务分

收稿日期: 2017-05-03; 修回日期: 2017-08-18.

责任编辑: 林崇.

作者简介: 滑楠(1974—), 男, 教授, 博士后, 从事动态任务分配等研究; 赵延龙(1992—), 男, 硕士生, 从事动态任务分配、群体智能的研究.

[†]通讯作者. E-mail: 1241492516@qq.com

配模型和基于TSPSO算法的模型求解方法进行仿真验证. 仿真结果表明, TSPSO算法能够快速寻找到机动通信保障任务最优分配方案.

1 机动通信保障任务分配的数学模型

1.1 决策变量设计

设有 M 个机动通信保障单元, 需要对 N 个通信地点进行保障, 保障地域中道路的类型数为 R . 在实际情况下, 安排一个保障单元对某个地点进行保障, 主要从保障效能、保障成本和保障时间3方面考虑, 决策变量设计为:

q_{mn} : 保障单元 m 对通信地点 n 的保障效能;

w_{mn} : 保障单元 m 对通信地点 n 的保障成本;

NT: 保障单元最多可保障的通信地点数量;

d_{mn}^i : 保障单元 m 与通信地点 n 间第 i 条可行路径的距离;

λ_{mnr}^i : 保障单元 m 与通信地点 n 间第 i 条可行路径中 r 类型道路的比例;

v_r : 保障装备在 r 类型道路上的平均运载速度.

1.2 优化目标及约束条件设计

机动通信保障任务分配问题主要从保障效能、保障成本和保障时间3个方面进行建模.

1.2.1 保障效能最优

保障效能最优是指在机动通信保障单元对通信地点的保障关系动态变化的情况下寻找确定的保障方案, 使得总体保障效能最优, 有

$$\max f_1 = \sum_{m=1}^M \sum_{n=1}^N x_{mn} q_{mn}, \quad (1)$$

其中 $x_{mn} \in \{0, 1\}$, 且 $x_{mn} = 1$ 表示保障单元 m 负责保障通信地点 n , 否则不保障. 一般情况下 $N > M$, 因此对于某一通信地点而言, 总是要有保障单元进行保障, 即 x_{mn} 需满足

$$\sum_{m=1}^M x_{mn} = 1, \quad n = 1, 2, \dots, N. \quad (2)$$

对于某一保障单元 m , 其保障能力是有限的, 因此保障单元 m 保障通信地点的总数不能大于某一具体值 (NT), 即需要满足

$$\sum_{n=1}^N x_{mn} \leq \text{NT}, \quad m = 1, 2, \dots, M. \quad (3)$$

1.2.2 保障成本最小

保障成本最小与保障效能类似, 即在同一保障方案下, 使得保障成本最小, 有

$$\min f_2 = \sum_{m=1}^M \sum_{n=1}^N x_{mn} w_{mn}; \quad (4)$$

$$\begin{aligned} \text{s.t. } & \sum_{m=1}^M x_{mn} = 1, \quad \sum_{n=1}^N x_{mn} \leq \text{NT}, \\ & m = 1, 2, \dots, M, \quad n = 1, 2, \dots, N, \quad x_{mn} \in \{0, 1\}. \end{aligned} \quad (5)$$

1.2.3 保障时间最短

保障时间最短是使得各个机动通信保障单元到不同通信地点保障时间的最大值最小, 有

$$\begin{aligned} \min f_3 = & \max_{m=1}^M \max_{n=1}^N \left(x_{mn} \min_i \left(\sum_r \frac{d_{mn}^i \lambda_{mnr}^i}{v_r} \right) \right), \\ & m = 1, 2, \dots, M, \quad n = 1, 2, \dots, N, \quad r = 1, 2, \dots, R. \end{aligned} \quad (6)$$

1.3 任务分配模型

综合考虑上述3方面的影响因素, 最终得到的机动通信保障任务分配模型为

$$\max F = \alpha f_1 - \beta f_2 - \gamma f_3; \quad (7)$$

$$\begin{aligned} \text{s.t. } & \sum_{m=1}^M x_{mn} = 1, \\ & \sum_{n=1}^N x_{mn} \leq \text{NT}. \end{aligned} \quad (8)$$

其中: $m = 1, 2, \dots, M, n = 1, 2, \dots, N, r = 1, 2, \dots, R$; α, β, γ 分别表示子目标 f_1, f_2, f_3 的权重.

2 二次搜索粒子群算法(TSPSO)

2.1 PSO算法原理

假设有 N 个粒子构成一个种群 S , X_i 表示 D 维搜索空间中的一个元素, 有

$$\begin{aligned} X_i = & (x_{i1}, x_{i2}, \dots, x_{iD}), \\ & i = 1, 2, \dots, N. \end{aligned} \quad (9)$$

V_i 表示第 i 个粒子的“飞行”速度, 有

$$\begin{aligned} V_i = & (v_{i1}, v_{i2}, \dots, v_{iD}), \\ & i = 1, 2, \dots, N. \end{aligned} \quad (10)$$

X_i 当前搜索到的最优适应度值所对应的位置称为个体极值, 有

$$\begin{aligned} P_i = & (p_{i1}, p_{i2}, \dots, p_{iD}), \\ & i = 1, 2, \dots, N. \end{aligned} \quad (11)$$

种群 S 当前搜索到的最优位置称为全局极值, 有

$$P_g = (p_{g1}, p_{g2}, \dots, p_{gD}). \quad (12)$$

所有粒子按照如下公式更新速度和位置信息:

$$\begin{aligned} v_{id} = & wv_{id} + c_1 r_1 (p_{id} - x_{id}) + \\ & c_2 r_2 (p_{gd} - x_{id}), \end{aligned} \quad (13)$$

$$x_{id} = x_{id} + v_{id}. \quad (14)$$

其中: w 为惯性权重; c_1 、 c_2 为学习速率; r_1 、 r_2 为 $[0,1]$ 区间内均匀分布的随机数。

2.2 TSPSO算法原理

PSO 中, 每个粒子由位置信息与速度信息组成, 所有粒子由统一的目标函数决定其相应的适应度值, 然后粒子追随当前个体及全局最优值进行搜索。

梯度下降法具有计算过程简单、初始收敛较快等优点^[13], 基本思想是某一质点沿着函数 $f(p)$ (p 为 D 维向量) 梯度下降方向可以快速滑落至函数的极值点处, 主要由两部分组成:

1) 计算搜索方向. 按照下式计算负梯度, 即最佳搜索方向:

$$d^k = -\nabla f(p^k) = -\left(\frac{\partial f(p^k)}{\partial p_1^k}, \frac{\partial f(p^k)}{\partial p_2^k}, \dots, \frac{\partial f(p^k)}{\partial p_D^k}\right). \quad (15)$$

2) 计算搜索步长. λ_k 取最优步长必须满足

$$f(p^k + \lambda_k d^k) = \min_{\lambda} f(p^k + \lambda d^k). \quad (16)$$

针对 PSO 算法中全局极值 p_g 在若干次迭代后保持不变, 造成粒子群体的多样性降低、陷入局部最优的缺陷, 在 TSPSO 算法中, 引入极值陷阱和禁忌区域两个定义。

定义 1 (极值陷阱) 对于给定的 D 维空间中某一曲面方程 $S(X) = 1$ (X 为 D 维向量), 其中非平坦区域 (梯度为非零向量) 中存在某一粒子 $X_0 \in R^D$, 当 X_0 可以朝着负梯度的方向迅速收敛至陷阱的谷点 (极值点) 时, X_0 所处的区域称为极值陷阱。

定义 2 (禁忌区域) 处于极值陷阱中的粒子朝着负梯度方向收敛到极值点, 为防止其他粒子重复搜索该“陷阱”, 增加时间开销而设定的以该极值点为中心、某一定长为“半径”的 D 维“圆域”空间称为禁忌区域。

TSPSO 算法的基本原理是, 当全局极值保持的次数超过最大预设值 N_{\max} 时, 与全局极值对应的粒子落入极值陷阱之中; 利用梯度下降法对该粒子进行二次搜索, 最终得到更优的全局极值 p'_g . 以 p'_g 为中心、某一确定长度 R 为“半径”构成“圆域” $\Omega_R(p'_g)$, 即禁忌区域. 当粒子 X_j ($j = 1, 2, \dots, N, j \neq i$) 搜索到 $\Omega_R(p'_g)$ 边界时, 令 V_j 反射, 从而防止粒子重复搜索 $\Omega_R(p'_g)$ 区域, 避免陷入局部最优, 提高寻优效率. 由于粒子 X_i 采用梯度下降法进一步寻优时, 最终落入谷点, 不再适合作为粒子群中的个体继续寻优, 该粒子将被淘汰, 此时依据粒子群多样性准则, 在搜索空间之内、禁忌区域 $\Omega_R(p'_g)$ 之外生成新的粒子替换已被淘汰的粒子。

2.3 TSPSO算法实现

TSPSO 算法的实现与标准 PSO 算法及其他改进 PSO 算法相比增加了判断极值陷阱、二次搜索寻优、设定禁忌区域、粒子淘汰与生成 4 个部分, 用来提高算法的搜索效率和寻优性能。

2.3.1 判断极值陷阱

由 PSO 算法参数设置^[14] 分析得到粒子搜索后期的寻优轨迹方程为

$$\lim_{t \rightarrow +\infty} X_i(t) = \frac{c_1 r_1 p_i + c_2 r_2 p_g}{c_1 r_1 + c_2 r_2}, \quad (17)$$

其中 r_1 、 r_2 为满足 $[0,1]$ 范围内均匀分布的随机变量. 对式 (17) 两边求期望, 得

$$\lim_{t \rightarrow +\infty} E(X_i(t)) = E\left(\frac{c_1 r_1 p_i + c_2 r_2 p_g}{c_1 r_1 + c_2 r_2}\right) = \frac{c_1 p_i}{c_1 + c_2} + \frac{c_2 p_g}{c_1 + c_2}. \quad (18)$$

由式 (18) 分析得出: 对于一般的粒子 X_i , 全局极值 p_g 处的适应度值要优于个体极值 p_i 处的适应度值, 因此粒子在寻优时受全局极值 p_g 的影响相对较大. 当粒子群经过若干次迭代寻优后, 全局极值 p_g 保持不变, 并且与 p_g 处的极值陷阱对应的全局极值 p'_g 并不是真正的全局最优值, 便会错误地引导其他粒子朝 p_g 方向移动, 导致越来越偏离真正的全局最优值, 最终陷入局部最优. 因此, 设定 PSO 在寻优过程中, 当全局极值保持不变的次数超过预设值时, 判断相应的粒子处于极值陷阱中。

2.3.2 二次搜索寻优

二次搜索是指当全局极值 p_g 处于极值陷阱中时, 为了防止 p_g 对其他粒子错误引导而导致局部最优, 需要对 p_g 进行进一步的独立寻优搜索过程. 本文选择梯度下降法进行二次搜索寻优以兼顾效率和性能。

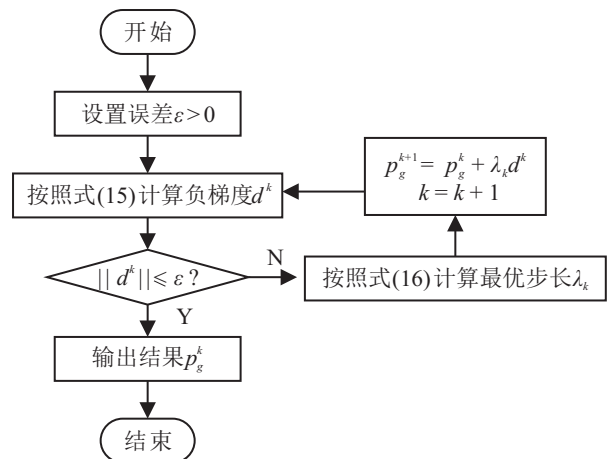


图 1 二次寻优流程

假设对每一个粒子都有一个固定的评价函数,即算法的适应度函数 $f(X_i)$,且函数 $f(\cdot)$ 在第 $i(i = 1, 2, \dots, D)$ 维的偏导数存在,那么二次寻优的流程如图1所示.

2.3.3 设定禁忌区域

禁忌区域是指以极值陷阱的谷点 p'_g 为中心、某一具体长度 R 为“半径”所构成的“圆域”空间 $\Omega_R(p'_g)$. 假设某一粒子 X_i 再次搜索到 $\Omega_R(p'_g)$ 边缘时,满足如下关系式:

$$\sqrt{\sum_{j=1}^D (x_{ij} - p'_{gj})^2} \leq R. \quad (19)$$

此时,为了防止粒子 X_i 重复搜索 $\Omega_R(p'_g)$ 区域,提高寻优的效率,需要对粒子的搜索方向进行反射操作,如图2所示. 其中:半径为 R 的圆域 $\Omega_R(p'_g)$ 为禁忌区域; V_i, V'_i 分别为粒子 X_i 的“入射”向量和“反射”向量.

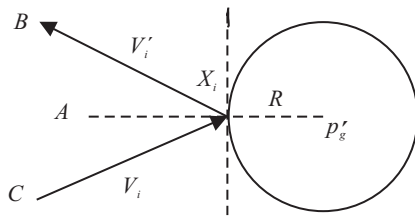


图2 粒子 X_i 反射示意

由光的反射定理可得

$$\begin{cases} \angle AX_i C = \angle AX_i B, \\ \|V_i\| = \|V'_i\|. \end{cases} \quad (20)$$

即向量 V_i 与向量 V'_i 的和向量与向量 $\overrightarrow{X_i p'_g}$ 垂直,通过余弦定理可以求得

$$\cos(\angle AX_i C) = \frac{\sum_{j=1}^D v_{ij}(x_{ij} - p'_{gj})}{2 \sqrt{\sum_{j=1}^D v_{ij}^2} \sqrt{\sum_{j=1}^D (x_{ij} - p'_{gj})^2}}, \quad (21)$$

$$V'_i = 2\|V_i\| \cos(\angle AX_i C) \frac{(X_i - p'_g)}{\|X_i - p'_g\|} + V_i. \quad (22)$$

将式(21)代入(22)得到 X_i 经反射后的方向向量 V'_i 为

$$V'_i = 2\|V_i\| \cos(\angle AX_i C) \frac{(X_i - p'_g)}{\|X_i - p'_g\|} + V_i = 2\sqrt{\sum_{j=1}^D v_{ij}^2} \times \frac{\sum_{j=1}^D v_{ij}(x_{ij} - p'_{gj})}{2\sqrt{\sum_{j=1}^D v_{ij}^2} \sqrt{\sum_{j=1}^D (x_{ij} - p'_{gj})^2}} \times$$

$$\frac{(X_i - p'_g)}{\sqrt{\sum_{j=1}^D (x_{ij} - p'_{gj})^2}} + V_i = \frac{\sum_{j=1}^D v_{ij}(x_{ij} - p'_{gj})}{\sum_{j=1}^D (x_{ij} - p'_{gj})^2} \times (X_i - p'_g) + V_i. \quad (23)$$

2.3.4 粒子淘汰与生成

当某一粒子 X_i 经过若干次迭代后进入极值陷阱中时,为防止该粒子对其他粒子的误导影响,需要对 X_i 进行单独的二次搜索寻优处理. 为防止局部最优,已不再适合接下来的寻优搜索过程,因此需将该粒子淘汰,并根据一定的准则生成新的粒子. 文献[15]中关于种群 S 多样性的计算函数式为

$$\text{Diversity}(S) = \frac{1}{N \cdot L_{\max}} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{ij} - \bar{z}_j)^2}. \quad (24)$$

其中: N 为粒子群的规模; D 为搜索空间的维度; L_{\max} 为 D 维搜索空间中的最大搜索半径; $\bar{z}_j(j = 1, 2, \dots, D)$ 为粒子群 S 在第 j 维的平均值. 由式(24)分析得出,当粒子的规模 N 取值较大时,个别粒子(如已淘汰的粒子)对 \bar{z}_j 的影响较小,甚至可以忽略. 若使新生成的粒子不影响整个粒子群的多样性,则必须满足 D 维向量 $(x_i - \bar{z})$ 的二范数值尽可能大. 因此,要求生成的粒子 X_i 必须尽可能地远离重心 \bar{z} , 即生成粒子越偏离 \bar{z} 其概率密度函数值越大,如下式所示:

$$f(x_{i1}, x_{i2}, \dots, x_{iD}) = \zeta \sum_{j=1}^D (x_{ij} - \bar{z}_j)^2. \quad (25)$$

其中 ζ 为常数,使得

$$\int \int \dots \int \zeta \sum_{j=1}^D (x_{ij} - \bar{z}_j)^2 dx_{i1} dx_{i2} \dots dx_{iD} = 1. \quad (26)$$

式(26)表示在搜索区域 S 的 D 重积分, S 即为该积分区域.

2.4 TSPSO算法流程

TSPSO算法的实现流程如下.

Step 1: 确定惯性权重 w , 学习速率 c_1, c_2 , 检测极值陷阱的最大迭代次数 MI 和粒子群体个数 N .

Step 2: 在给定的范围内,随机生成 N 个粒子位置 x_i , 速度 $v_i(i = 1, 2, \dots, N)$ 信息.

Step 3: 计算 x_i 的适应度值 f_i , 经判断若满足终止条件,则输出结果,否则继续.

Step 4: 更新个体极值 p_i 和全局极值 p_g , 由式 (13) 和 (14) 更新粒子速度和位置信息.

Step 5: 判断全局极值 p_g 保持不变的次数是否超过预设最大值, 满足则转至 Step 6, 否则返回 Step 3.

Step 6: 判断全局极值 p_g 是否处于极值陷阱中, 采用梯度下降法进行二次搜索寻优, 并用寻优结果 p'_g 更新 $p_g (p_g = p'_g)$.

Step 7: 将与全局极值 p_g 对应的粒子淘汰, 并依据种群多样性准则生成新粒子进行替换, 返回 Step 3.

TSPSO 算法流程如图 3 所示.

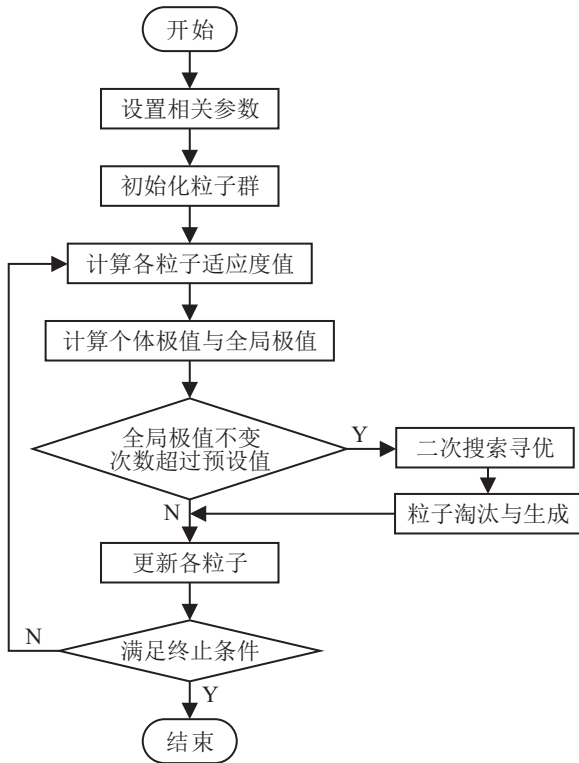


图 3 TSPSO 算法流程

2.5 TSPSO 算法测试分析

2.5.1 测试环境

软件环境: Microsoft Windows 7 操作系统, Matlab 2016a 仿真平台.

硬件环境: Intel core i5-3470, 主频 3.20 GHz 处理器, 4 G 内存, 联想机型.

本文对 4 种改进粒子群算法进行对照仿真测试: 线性递减权重粒子群算法^[16](Linearly decreasing weight particle swarm optimization, LDWPSO)、杂交粒子群算法^[17](Hybrid particle swarm optimization, HPSO)、自然选择粒子群算法^[18](Natural selection particle swarm optimization, NSPSO)、免疫粒子群算法^[19](Immune particle swarm optimization, IPSO). 选取文献 [20] 中 4 种典型的测试函数 F_1 (Sphere)、 F_2 (Rosenbrock)、 F_3 (Rastrigrin)、 F_4 (Griewank) 进行仿真实验.

2.5.2 算法参数设置

TSPSO 算法中, 最大迭代次数 MI 取值 10; LDWPSO 算法中, 惯性权重 w 变化范围 [0.2, 0.8]; HPSO 算法中, 杂交概率 bc 取值 0.8, 杂交池的大小比例 bs 取值 0.1; IPSO 算法中, 检测免疫的迭代次数 DS 取值 8, 免疫替换概率 $p_{replace}$ 取值 0.5. 改进 PSO 算法其余的惯性权重 w 、认知项权重 c_1 和社会项权重 c_2 分别取值 0.8、2 和 2; 粒子个数 N 取值 50; 迭代次数 M 分别取值 1 000、2 000 和 3 000; 搜索空间维数 D 分别取值 10、20 和 30.

2.5.3 测试结果分析

对 5 种改进 PSO 算法进行测试, 所得结果如表 1 和图 4 所示.

表 1 5 种 PSO 算法对 4 种测试函数仿真结果

函数	维数 D	迭代次数 M	函数的平均最小值 (运行 10 次)				
			TSPSO	LDWPSO	HPSO	NSPSO	IPSO
F_1	10	1 000	2.36e-08	1.61e-07	1.45e-07	7.24e-06	4.47e-06
	20	3 000	1.09e-07	1.419 1	0.002 5	1.24e-04	4.26e-04
	30	3 000	3.48e-07	13.934 0	0. 202 2	0.213 4	0.086 7
F_2	10	1 000	3.62e-04	5.286 9	4.486 2	4.107 6	3.648 9
	20	2 000	1.15e-04	164.391 3	17.542 7	27.763 7	18.404 8
	30	3 000	0.018 7	2.62e+03	29.257 3	32.689 6	28.626 9
F_3	10	1 000	3.04e-09	33.968 8	28.579 1	24.332 0	25.431 0
	20	2 000	2.67e-08	144.384	69.408	68.767	68.348
	30	3 000	6.12e-07	258.139	138.051	201.239	139.703
F_4	10	3 000	1.39e-07	0.132 5	0.167 8	0.156 7	0.162 4
	20	3 000	6.81e-06	0.269 0	0.395 8	0.418 1	0.340 1
	30	3 000	1.32e-05	1.117 2	1.092 3	1.066 6	1.126 2

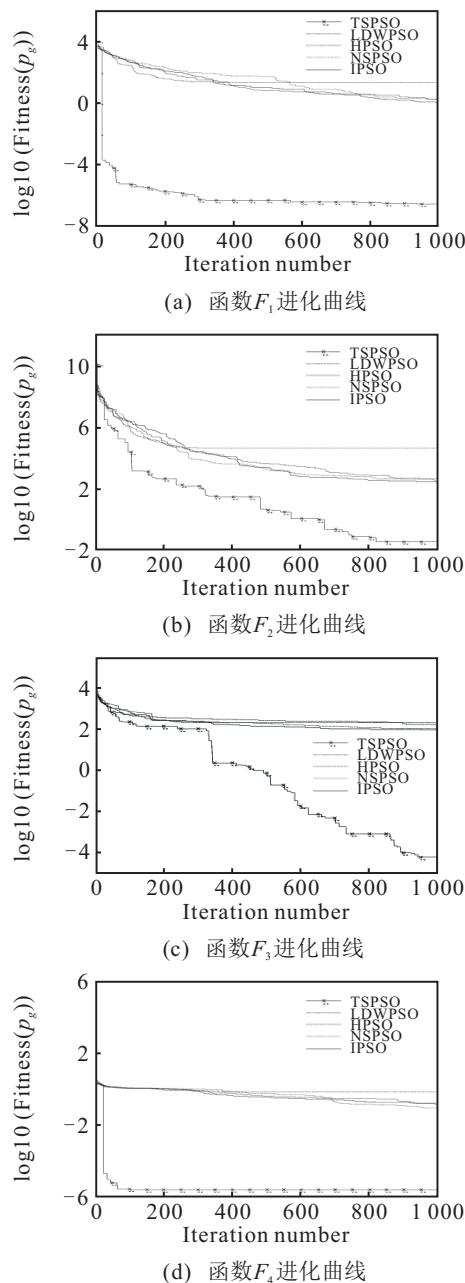


图4 5种PSO算法对4种测试函数的全局极值进化曲线

由表1可见:TSPSO算法在4种测试函数上的收敛精度均优于LDWPSO、HPSO、NSPSO、IPSO算法,且对于某一测试函数(尤其 F_2),随着空间维度的增加,TSPSO算法的收敛精度保持相对稳定,而其他算法收敛精度则急剧降低。

图4的测试函数维度均为30,5种PSO算法迭代次数均为1000.可见,TSPSO算法的收敛速度和精度均优于4种对照PSO算法。

针对不同PSO算法对同一测试函数收敛到相同精度的花费时长进行比较,结果如表2示。

由表2可见:函数 F_1 在不同维度且收敛到相同精度的条件下,TSPSO算法与其他4种改进PSO算法相比花费时长更短;函数 F_2 在维度为20和30且收敛到相同精度的条件下,TSPSO算法与其他4种改进PSO算法相比花费时长更短;函数 F_3 在维度为30且收敛到相同精度的条件下,TSPSO算法与其他4种改进PSO算法相比花费时长更短;函数 F_4 在维度为10且收敛到相同精度的条件下,TSPSO算法与其他4种改进PSO算法相比花费时长更短.对于其他情况,TSPSO算法的花费时长与其他算法相比虽然不是最短,但相差很小,从算法的收敛精度方面考虑可以忽略,因此TSPSO算法的寻优效率总体上较优。

3 基于TSPSO算法的任务分配模型求解

3.1 粒子编码与解码

本文所需解决的任务分配问题主要是计算 M 个机动通信保障单元对 N 个通信地点的保障方案问题,即需要通过计算求解出一个 $M \times N$ 规模的分配矩阵 X .由于粒子中所存储的信息通常是一维实数向量信息,本文采用长度为 $M \times N$ 的一维向量形式 $(p_{11}, p_{12}, \dots, p_{M(N-1)}, p_{MN})$ 代表一个粒子 x ,其中

表2 5种PSO算法对4种测试函数花费时长比较

函数	维数 D	收敛精度	函数的平均最小值(运行10次)				
			TSPSO	LDWPSO	HPSO	NSPSO	IPSO
F_1	10	e-06	0.042 978	0.048 851	0.467 731	0.087 836	0.096 264
	20	1	0.045 837	0.949 130	0.664 380	0.174 772	0.157 407
	30	10	0.048 011	1.223 726	1.217 637	0.386 563	0.394 731
F_2	10	1	1.036 785	0.186 526	0.328 429	0.233 219	0.102 303
	20	100	0.345 702	0.860 188	2.366 042	0.589 509	0.639 260
	30	1000	0.357 433	0.933 664	2.608 548	0.773 426	0.794 290
F_3	10	10	0.022 974	0.029 919	0.050 699	0.021 837	0.024 81
	20	100	0.013 092	0.017 064	0.019 039	0.011 129	0.011 985
	30	100	0.046 724	0.059 255	0.217 939	0.076 309	0.054 182
F_4	10	e-01	0.002 905	0.006 908	0.019 920	0.007 596	0.007 334
	20	e-01	0.175 732	0.175 732	0.221 810	0.140 098	0.131 216
	30	1	0.002 537	0.001 924	0.002 551	0.001 951	0.001 841

$p_{mn} \in [0, 1]$ 表示保障单元 m 对通信地点 n 的保障概率分配情况. 例如 $p_{11}, p_{21}, \dots, p_{M1}$ 分别表示保障单元 $1, 2, \dots, M$ 对通信地点 $n(n = 1, 2, \dots, N)$ 的保障概率大小, 因此满足

$$\sum_{m=1}^M p_{mn} = 1, n = 1, 2, \dots, N. \quad (27)$$

因为对于某个保障单元 m 的保障能力是有限的, 即保障单元 m 同时保障的通信地点总数不能超过某一具体值 (NT), 因此需满足

$$\sum_{n=1}^N p_{mn} \leq NT, m = 1, 2, \dots, M. \quad (28)$$

解码时, $p_{11}, p_{21}, \dots, p_{M1}$ 中最大概率值的序号 $m_{\max} \in \{1, 2, \dots, M\}$ 表示保障单元 m_{\max} 保障通信地点 1, 即

$$x_{m1} = \begin{cases} 1, & m = m_{\max}; \\ 0, & \text{Otherwise.} \end{cases} \quad (29)$$

3.2 适应度函数

本文建立的机动通信保障任务分配模型综合考虑保障效能最优、保障成本最小和保障时间最短三方面因素, 因此在确定适应度函数时同样需要考虑这几方面因素. 由式 (7) 得出: 当任务分配方案越优时, 保障效能 f_1 越高, 保障成本 f_2 越少, 保障时间 f_3 越短, 总体的任务分配模型指标 F 越大. 适应度函数为

$$f(x^*) = \frac{1}{\alpha f_1 - \beta f_2 - \gamma f_3 + C};$$

$$\text{s.t. } f_1 = \sum_{m=1}^M \sum_{n=1}^N x_{mn} q_{mn},$$

$$f_2 = \sum_{m=1}^M \sum_{n=1}^N x_{mn} w_{mn},$$

$$f_3 = \max_{m=1}^M \max_{n=1}^N \left(x_{mn} \min_i \left(\sum_r \frac{d_{mn}^i \lambda_{mnr}^i}{v_r} \right) \right). \quad (30)$$

其中: x^* 为一个可行的粒子; α, β, γ 分别为优化子目标 f_1, f_2, f_3 的相对权值, 且满足 $\alpha + \beta + \gamma = 1$; C 为固定常数, 用来防止式 (30) 分母取为零的情况.

对于合法的粒子 $x^* = (x_{11}, x_{12}, \dots, x_{M(N-1)}, x_{MN})$, 必须满足式 (5) 的约束条件.

4 仿真分析

为验证本文所建立机动通信保障任务分模型的合理性和基于二次搜索的改进 PSO 算法对模型求解的可靠性, 在第 2.5.1 节测试环境下对具体实例进行仿真验证.

假设 4 个保障单元 Unit₁、Unit₂、Unit₃、Unit₄, 需要保障 6 个通信地点 Point₁、Point₂、 \dots 、Point₆. 各保障单元对不同通信地点的保障效能、保障成本和保障时间如表 3 所示.

表 3 各保障单元对不同通信地点保障效能、保障成本和保障时间

保障单元	通信地点					
	Point ₁	Point ₂	Point ₃	Point ₄	Point ₅	Point ₆
Unit ₁	0.8147、0.6787、0.7981	0.6324、0.6555、1.2285	0.9575、0.2769、0.9123	0.9572、0.6948、1.0716	0.4218、0.4387、0.9848	0.6557、0.1869、1.2498
Unit ₂	0.9058、0.7577、1.4030	0.0975、0.1712、1.5125	0.9649、0.0462、1.3694	0.4854、0.3171、0.9104	0.9157、0.3816、1.0897	0.0357、0.4898、1.2759
Unit ₃	0.1270、0.7431、0.9442	0.2785、0.7060、1.1955	0.1576、0.0917、0.8307	0.8003、0.9502、1.0368	0.7922、0.7655、0.9438	0.8491、0.2456、0.8871
Unit ₄	0.9134、0.3922、0.9602	0.5469、0.0318、0.9534	0.9706、0.8235、0.9669	0.1419、0.0344、1.0206	0.9595、0.7952、1.2732	0.9340、0.6463、0.9133

改进 PSO 算法中, 惯性权重 w 、认知项权重 c_1 和社会项权重 c_2 分别取值 0.8、0.2 和 0.2; 粒子规模 N 与迭代次数 M 分别取值 20 和 200; 检测极值陷阱的最大迭代次数 MI 取值 10; 适应度函数中 3 个分目标 f_1, f_2, f_3 的权重 α, β, γ 分别取值 0.5、0.3、0.2, 常数 C 取值 24. 通过 Matlab 2016a 软件进行仿真, 由于

标准 PSO 以及改进 PSO 算法的初始粒子及相应速度是在一定范围内随机生成, 对最终的优化结果有一定影响, 本文进行 10 次仿真实验, 得到相对较好的仿真结果如表 4 所示.

将表 4 中各保障单元对不同通信地点的保障概率按照第 3.1 节中粒子的解码方法解码为最终保障

方案,如表5所示.

表4 各保障单元对不同通信地点保障概率

保障单元	通信地点					
	Point ₁	Point ₂	Point ₃	Point ₄	Point ₅	Point ₆
Unit ₁	0.4887	8.05e-6	0.8863	0.9458	9.55e-6	2.22e-5
Unit ₂	0.0275	1.58e-6	0.0891	0.0542	0.9999	0.0299
Unit ₃	8.42e-6	2.51e-5	0.0246	2.47e-5	4.42e-6	0.9701
Unit ₄	0.4838	0.9999	3.98e-6	5.05e-6	1.32e-5	2.71e-6

表5 各保障单元对不同通信地点保障方案

保障单元	通信地点					
	Point ₁	Point ₂	Point ₃	Point ₄	Point ₅	Point ₆
Unit ₁	0	0	1	1	0	0
Unit ₂	0	0	0	0	1	0
Unit ₃	0	0	0	0	0	1
Unit ₄	1	1	0	0	0	0

从表5中分析得出具体保障方案为:保障单元Unit₁保障通信地点Point₃、Point₄;保障单元Unit₂保障通信地点Point₅;保障单元Unit₃保障通信地点Point₆;保障单元Unit₄保障通信地点Point₁、Point₂.将保障方案代入式(30)计算适应度值为0.0403.

4.1 保障方案合理性分析

如表5的保障方案:对于通信地点Point₁,虽然保障单元Unit₄的保障时间并不是最短,但与最短的保障单元Unit₁的保障时间相差较小,并且在4个保障单元中保障效能最优、保障成本最小,综合三方面因素,通信地点Point₁由保障单元Unit₄进行保障比较合理;对于通信地点Point₂,保障单元Unit₄的保障效能排在4个保障单元中的第2位,与第1位Unit₁的保障效能仅相差0.0855,然而在4个保障单元中,保障单元Unit₄的保障成本最低且保障时间最短,因此通信地点Point₂由保障单元Unit₄进行保障比较合理;对于通信地点Point₃,保障单元Unit₂的保障时间最长且与其他保障单元相差较大,保障单元Unit₃的保障效能最差且与其他保障单元差距较大,保障单元Unit₄的保障成本最高且与其他保障单元相差较大,因此通信地点Point₃由保障单元Unit₁进行保障比较合理;对于通信地点Point₄,保障单元Unit₁的保障效能最优,并且保障成本较低、保障时间较短,因此通信地点Point₄由保障单元Unit₁进行保障比较合理;对于通信地点Point₅,保障单元Unit₂的保障成本最低,并且保障效能相对较优、保障时间较短,因此通信

地点Point₅由保障单元Unit₂进行保障比较合理;对于通信地点Point₆,保障单元Unit₃的保障时间最短,并且保障效能相对较优、保障成本较低,因此通信地点Point₆由保障单元Unit₃进行保障比较合理.

4.2 保障方案可行性分析

为验证表5中机动通信保障任务分配方案的可行性,即在当前任务分配模型和仿真环境条件下所求保障方案是否最优,利用Lingo16.0软件进行编程求解.当任务分配模型的数据规模较大时,Lingo16.0软件求解需花费大量时间,甚至得不到全局最优值,不能满足复杂环境下实时任务分配的需求,因此仅利用Lingo16.0对4种具体实例进行验证,最终结果与表5中的保障方案完全吻合,且全局最优值与4种计算的适应度值相等,从而验证了保障方案的可行性.

5 结论

本文通过分析机动通信保障任务分配问题的实际制约因素,建立了机动通信保障任务分配的数学模型.在标准PSO算法的基础上增加判断极值陷阱、二次搜索寻优、设定禁忌区域、粒子淘汰与生成4个部分,并结合梯度下降算法的快速寻优特点,提出一种基于二次搜索的改进粒子群优化算法(TSPSO).在基于TSPSO算法的任务分配模型求解方法中,利用保障单元到通信地点分配概率的思想对粒子进行编码与解码,提高任务分配模型的求解效率.最后通过机动通信保障任务分配具体实例验证了所建立模型的科学性,以及改进PSO算法对求解任务分配保障方案的合理性和可行性.

参考文献(References)

- [1] 夏军,庞征斌,张峻,等.一种基于0-1整数规划的全局数据分布优化方法[J].国防科技大学学报,2009,31(4):62-67.
(Xia J, Pang Z B, Zhang J, et al. A 0-1 integer programming based approach for global data distribution[J]. J of National University of Defense Technology, 2009, 31(4): 62-67.)
- [2] 李宏,焦永昌,张莉.一种求解混合整数规划的混合进化算法[J].控制与决策,2008,23(10):1098-1102.
(Li H, Jiao Y C, Zhang L. Hybrid evolutionary algorithm for mixed-integer programming problems[J]. Control and Decision, 2008, 23(10): 1098-1102.)
- [3] 王虹,卫军胡,刘昌军,等.引入群体发现和加入行为的随机搜索算法[J].西安交通大学学报,2013,47(12):43-49.
(Wang H, Wei J H, Liu C J, et al. A novel adaptive stochastic search algorithm based on group founding and

- joining behaviors[J]. J of Xi'an Jiaotong University, 2013, 47(12): 43-49.)
- [4] Jia L P, Wang Y P, Fan L. An improved uniform design-based genetic algorithm for multi-objective bilevel convex programming[J]. Int J Computation Science and Engineering, 2016, 12(1): 38-46.
- [5] Ben M, Patricia R, Cameron S, et al. Evaluating the seeding genetic algorithm[J]. Lecture Notes in Computer Science, 2013, 8272(1): 228-233.
- [6] 方璐, 詹军, 徐先勇, 等. 基于多智能体遗传算法的配电网节能降耗综合管理系统[J]. 湖南大学学报: 自然科学版, 2016, 43(4): 105-112.
(Fang L, Zhan J, Xu X Y, et al. Energy saving management system of distribution network based on multi-agent genetic algorithm[J]. J of Hu'nan University: Natural Science, 2016, 43(4): 105-112.)
- [7] Liao T J, Thomas S. A unified ant colony optimization algorithm for continuous optimization[J]. European J of Operational Research, 2014, 234(3): 597-609.
- [8] 马荣贵, 崔华, 薛世焦, 等. 改进蚁群算法的多约束质量最优路径选择[J]. 西安电子科技大学学报: 自然科学版, 2016, 43(3): 185-189.
(Ma R G, Cui H, Xue S J, et al. Improved ant colony algorithm for the optimal-quality-path routing problem with multi-constraints[J]. J of Xidian University: Natural Science, 2016, 43(3): 185-189.)
- [9] 王生生, 杨娟娟, 柴胜. 基于混沌鲶鱼效应的人工蜂群算法及应用[J]. 电子学报, 2014, 42(9): 1731-1737.
(Wang S S, Yang J J, Chai S. Artificial bee colony algorithm with chaotic catfish effect and its application[J]. Acta Electronica Sinica, 2014, 42(9): 1731-1737.)
- [10] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of the 4th Int Conf on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [11] 罗磊, 陈恳, 杜峰坡, 等. 基于改进型粒子群算法的曲面匹配与位姿获取[J]. 清华大学学报: 自然科学版, 2015, 55(10): 1061-1066.
(Luo L, Chen K, Du F P, et al. Surface fitting and position-pose measurements based on an improved SA-PSO algorithm[J]. J of Tsinghua University: Science and Technology, 2015, 55(10): 1061-1066.)
- [12] 严浙平, 邓超, 赵玉飞, 等. 改进粒子群算法在UUV航迹规划中的应用[J]. 华中科技大学学报: 自然科学版, 2013, 41(12): 64-68.
(Yan Z P, Deng C, Zhao Y F, et al. Application of improved particle swarm algorithm to UUV 3-D path planning[J]. J of Huazhong University of Science and Technology: Natural Science Edition. 2013, 41(12): 64-68.)
- [13] 许少华, 宋美玲, 许辰, 等. 一种基于混合误差梯度下降算法的过程神经网络训练[J]. 东北石油大学学报, 2014, 38(4): 92-96.
(Xu S H, Song M L, Xu C, et al. Training algorithm of process neural networks based on hybrid error gradient descent[J]. J of Northeast Petroleum University, 2014, 38(4): 92-96.)
- [14] 汤可宗, 李慧颖, 李娟, 等. 一种求解复杂优化问题的改进粒子群优化算法[J]. 南京理工大学学报, 2015, 39(4): 386-391.
(Tang K Z, Li H Y, Li J, et al. Improved particle swarm optimization algorithm for solving complex optimization problems[J]. J of Nanjing University of Science and Technology, 2015, 39(4): 386-391.)
- [15] Riget R, Vesterstrom J S. A diversity-guided particle swarm optimizer-he ARPSO[R]. Denmark: University of Aarhus, 2002.
- [16] 汤可宗. 遗传算法与粒子群优化算法的改进及应用研究[D]. 南京: 南京理工大学计算机科学与工程学院, 2012.
(Tang K Z. Improvement and application of genetic algorithm and particle swarm algorithm research[D]. Nanjing: Institute of Computer Science and Engineering, Nanjing University of Technology, 2012.)
- [17] 周利军, 彭卫, 曾小强, 等. 基于杂交变异的动态粒子群优化算法[J]. 计算机科学, 2013, 40(11A): 143-146.
(Zhou L J, Peng W, Zeng X Q, et al. Dynamic particle swarm optimization based on hybrid variable[J]. Computer Science, 2013, 40(11A): 143-146.)
- [18] 白俊强, 尹戈玲, 孙智伟, 等. 基于二阶振荡及自然选择的随机权重混合粒子群算法[J]. 控制与决策, 2012, 27(10): 1459-1465.
(Bai J Q, Yin G L, Sun Z W, et al. Random weighted hybrid particle swarm optimization algorithm based on second order oscillation and natural selection[J]. Control and Decision, 2012, 27(10): 1459-1465.)
- [19] Wei J X, Sun Y H, Su X N. A novel particle swarm optimization based on immune selection[J]. J of Nanjing University: Natural Sciences, 2010, 46(1): 1-8.
- [20] Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009, 39(6): 1362-1381.

(责任编辑: 郑晓蕾)