非线性 TP 的 PSO 求解*)

张美玉[1] 黄 翰[2] 郝志峰[2]
(解放军信息工程大学电子技术学院广州训练大队 广州 510510)[1]
(华南理工大学计算机科学与工程学院 广州 510640)[2]

摘 要 运输问题自提出后,人们因其在各个领域的广泛应用进行了大量研究。尤其是线型运输问题,已经设计出了多种有效解法,但它们均不能直接处理非线性运输问题。本文在经典粒子群算法 PSO 的基础上设计了新算法 PSO-NLTP,它通过改进 PSO 的粒子飞行速度和飞行位置更新方程,及设计出负修复算子,既满足 TP 的约束条件,又扩大了搜索空间。针对经典 PSO 算法容易在局部最优解过早停止搜索的不足,我们添加了自适应的变异算子,以防止 PSO-NLTP 过早停止搜索。通过仿真实例证明,与遗传算法 GA-NLTP 和带惩罚策略的 EP 进行比较,PSO-NLTP 能在较短的时间内找到更优解,结果验证了新算法的有效性。

关键词 非线性运输问题,粒子群算法,负修复,自适应变异

Particle Swarm Optimization Algorithm for Solving Non-linear Transportation Problem

ZHANG Mei-yu[1] HUANG Han[2] HAO Zhi-feng[2]
(Institute of Electronic Technology, the PLA Information Engineering University, Guangzhou 510510, China)[1]
(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)[2]

Abstract The transportation problem (TP) is well known as a combinatorial optimization problem for it could be extensively applied in many fields. There are several mathematical and computational methods for linear transportation problem (LTP). However, the approaches cannot be used in solving non-linear transportation problems (NLTP) directly. In the present paper, a new algorithm (PSO-NLTP) is proposed for the solution to NLTP on the PSO algorithm. Taking use of the function of the new position updating rule and negative repair operator, it can satisfy the constrained conditions of TP. PSO-mutation as another extra operator is introduced to enlarge the search space. The nature of PSO can accelerate the convergence of the novel algorithm, which would also make PSO-TP get the local best solution. However, the PSO mutation as an extra operator can help PSO-TP to avoid mature convergence. The numerical experiments of solving the NLTP instances based on open problems show the effectiveness and efficiency of PSO-NLTP through the comparison with EP with penalty strategy and GA in the literature.

Keywords Non-linear transportation problem, Particle swarm optimization, Negative repair, Mutation

1 引言

运输问题(Transportation problem, TP)作为运筹学中的一个经典问题,也是一类特殊的线性规划问题。它不但有强烈的实际背景,而且有着广泛的理论应用,很多组合最优化问题与它息息相关(例如它与最小费用流问题可相互转化)。运输问题自 1941 年 Hichcock[1] 和 Kantorovic[2] 提出后,又逐渐发展了多种模型和多种解法[3]。如根据目标类型可将 TP 分为:

(1)线性问题或非线性问题;

(2)单目标问题或多目标问题。根据约束类型又可分为:①二维问题或三维问题;②平衡问题或非平衡问题。

根据约束条件的特殊结构,Dantzig[4] 首先提出了 the primal simplex algorithm (PSA) 求解 TP。在此基础之上改进的算法,如 BFT, SSF and AKP[5-8] 主要用来解决线性规划问题。其中 AKP 是最快的,但它需要有效的初始策略来求得初始解。这几种算法均不能处理非线性目标函数。不同于以上算法,M. Gen 等人[9-11] 在标准 GA 的基础上为 TPs 设计了一种遗传算法,结果显示算法行之有效。随后,Then Yang 和 Gen 发展了 TP 的 GA 并用于求解其它类型的 TP。但因为 TP 数据结构及操作算子的复杂性,算法需要大量的计算时间,而且只能用于求解整数 TP,无法直接求解实数问题。

本文改进了经典 PSO 算法以求解非线性 TP。为满足 TP 的约束条件(3)、(4),修改了标准的粒子群迭代方程;为满足约束条件(5),每代进化时对 PSO 添加了一个新的负修复算子。同时,为了扩大搜索空间,还引进了变异算子。仿真实例显示,和 GA、带惩罚策略的 EP 比较,PSO 算法更为快速和有效。

研究非线性的运输情况在理论和实践上都具有广泛意

*)国家自然科学基金(10471045, 60433020)、国家新世纪优秀人才基金(NCET-05-0734)、广东省自然科学基金(04020079)、霍英东基金(91005)、教育人文社科基金(2005-241)、广东省科技攻关项目(2005B10101010)。张美玉 讲师,研究方向为组合优化、仿生算法;黄 翰 博士研究生,研究方向为进化计算工具的理论基础、进化算法的改进设计研究、进化算法的应用;郝志峰 教授,博士生导师,研究方向为组合优化、仿生算法、仿生算法的数学基础。

义，如铁路运输问题[14-20]就是一个典型的非线性 TP。通过分析这一铁路运输实例说明，非线性的运输模型在现实中是普遍而广泛地存在的，这类问题的求解也亟待进一步地研究。

2 粒子群算法(PSO)

粒子群优化(Particle Swarm Optimization，PSO)算法是一种基于群智能方法的演化计算技术，最早是由 Kennedy 和 Eberhart 提出的[21,22]。同遗传算法类似，它是一种基于群体的仿生优化工具，其基本算法内容如下：

假设在一个 D 维的目标搜索空间中，有 m 个粒子组成一个群落，其中第 i 个粒子表示为一个 D 维的向量 $\vec{x}_i=(x_{i1},x_{i2},\cdots,x_{iD})$，$i=1,2,\cdots,m$ 即第 i 个粒子在 D 维的搜索空间中的位置是 $\vec{x}_i$。换言之，每个粒子的位置就是一个潜在的解。将 $\vec{x}_i$ 带入一个目标函数，就可以计算出其适应值，根据适应值的大小衡量 $\vec{x}_i$ 的优劣。第 i 个粒子的"飞翔"速度也是一个 D 维的向量，记为 $\vec{x}_i=(v_{i1},v_{i2},\cdots,v_{iD})$。记第 i 个粒子迄今为止搜索到的最优位置为 $\vec{p}_i=(p_{i1},p_{i2},\cdots,p_{iD})$，整个粒子群迄今为止搜索到的最优位置为 $\vec{p}_g=(p_{i1},p_{i2},\cdots,p_{iD})$。

Kennedy 和 Eberhart 最早提出的 PSO 算法采用下列公式对粒子操作，分别对粒子的飞行速度和飞行位置进行更新。

$$v_{id}=v_{id}+c_1r_1(p_{id}-x_{id})+c_2r_2(p_{gd}-x_{gd})\quad x_{id}=x_{id}+v_{id} \tag{1}$$

其中，$i=1,2,\cdots,m$，$d=1,2,\cdots,D$；学习因子 c_1 和 c_2 是非负常数；r_1 和 r_2 是介于[0 , 1]之间的随机数。$v_{id}\in[-v_{max},v_{max}]$，$v_{max}$是常数，由用户设定。

迭代终止条件根据具体问题一般选为最大迭代次数或(和)粒子群迄今为止搜索到的最优位置满足预定最小适应阈值。

3 PSO 求解非线性 TP

我们可以将研究的非线性 TP(Non-linear Transportation Problem，NLTP)归为如下模型：

$$\min\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}(x_{ij}) \tag{2}$$

$$\text{s.t.}\sum_{j=1}^{n}x_{ij}=a_i,i=1,2,\cdots,m \tag{3}$$

$$\sum_{i=1}^{m}x_{ij}=b_j,j=1,2,,\cdots,n \tag{4}$$

$$x_{ij}\geqslant 0,i=1,2,\cdots,m\ j=1,2,\cdots,n \tag{5}$$

以平衡问题为例，添加平衡条件

$$\sum_{i=1}^{n}a_i=\sum_{j=1}^{m}b_j \tag{6}$$

这里 $f_{ij}(x_{ij})\neq\cos t_{ij}\cdot x_{ij}$，是非线性的目标函数。例如，考虑到运输问题复杂的背景，目标函数可以为下列非线性函数[23]

非线性函数 1：$f_{ij}(x)=c_{ij}x_{ij}^2$ (7)

非线性函数 2：$f_{ij}(x)=c_{ij}\sqrt{x_{ij}}$ (8)

非线性函数 3：

$$f_{ij}(x)=\begin{cases}c_{ij}(\dfrac{x_{ij}}{S}), & \text{if } 0\leqslant x_{ij}<S\\ c_{ij}, & \text{if } S<x_{ij}\leqslant 2S\\ c_{ij}(1+\dfrac{x_{ij}-2S}{S}), & \text{if } 2S<x_{ij}\end{cases} \tag{9}$$

其中，S 是典型 x 值的阶。

非线性函数 4：$f_{ij}(x)=c_{ij}x_{ij}[\sin(x_{ij}\frac{5\pi}{4S})+1]$ (10)

本文设计的 PSO-NLTP 新算法根据非线性 TP 的特点，在原有 PSO 算法的基础上，改进了飞行位置和飞行速度更新方程，并添加了负修复算子。PSO-NLTP 的核心部分设计为：初始化，改进位置更新，负修复，PSO 变异。具体介绍如下。

3.1 PSO-NLTP 的初始化

在平衡假设条件(6)下，运输问题通常有可行解。对于不平衡 TP 一般可以转化为平衡 TP 来求解[24]。如一个非平衡的 $n\times m$ 的 TP 可以转化为一个 $n\times(m+1)$的平衡 TP 来解，第 $m+1$ 个终点的需求量在非平衡条件 $\sum_{i=1}^{n}a_i\geqslant\sum_{j=1}^{m}b_j$ 下可用 $b_{m+1}=\sum_{i=1}^{n}a_i-\sum_{j=1}^{m}b_j$ 来计算。

对于平衡 TP 通过以下的过程来获得初始解。

PSO-NLTP 的初始化设计如下：

```
Procedure Initialization
变量定义：i (整数);
begin
    获得平衡 NLTP;
    i:= 1;
    repeat
    CreateOneSolution (x_i,i);
    i:= i+1;
  until i>n*m
end.
```

其中辅助算子 CreateOneSolution 算法设计如下：

```
Procedure CreateOneSolution (变量 X:粒子, first: 整数)
变量定义：i,j,k (整数), N(整数集);
begin
    k :=0; N := {1,2,…,nm};
    repeat
        如果 k=0 那么 k:=first 否则 k:= N 中的随机元素;
        i :=⌊(k-1)/m+1⌋;j := {k-1} mod m+1;
        x_ij := min {a_i,b_j};a_i:=a_i-x_ij;b_j:=b_j-x_ij;N:= N \
{k};
    until N 取空
end
```

一个粒子 $x=\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1m}\\ x_{21} & x_{23} & \cdots & x_{2m}\\ \cdots & \cdots & \cdots & \cdots\\ x_{n1} & x_{m2} & \cdots & x_{nn}\end{bmatrix}$ 代表 NLTP 的一个解。在初始化过程中获得了 nm 个初始解。集合 N 中的每一个元素都能独立作为初次分配来获得解，这有利于种群在进化中获得最优解。

3.2 改进更新方程

在 PSO 算法中，问题能通过更新方程(1)来获得新解。但对于 TP，这个更新方程不能满足 TP 的约束条件(3)和(4)。

为此，本文设计了新的更新原则来克服这个不足。对于粒子 d 的位置 $X^d(n\times m)$能按下面的方程更新：

$$V_{t+1}^d=\begin{cases}\varphi_1(P_t^d-X_t^d)+\varphi_2(P_t^g-X_t^d), & \text{if } t=0\\ \lambda_1V_t^d+\lambda_2[\varphi_1(P_t^d-X_t^d)+\varphi_2(P_t^g-X_t^d)], & \text{otherwise}\end{cases}$$

$$X_{t+1}^d=V_{t+1}^d+X_t^d \tag{11}$$

$p_t^d(n\times m)$是粒子 d 在第 t 代当前最好的位置，$p_t^g(n\times m)$是粒子群在第 t 带全局最好位置。φ_1 和 φ_2 是位于(0 ,1)之间的随机数，满足 $\varphi_1+\varphi_2=1$。λ_1 是位于(0.8 , 1)之间的随机数，$\lambda_2=1-\lambda_1$。

如果 $t=0$，

$$\begin{aligned}X_{t+1}^d&=V_{t+1}^d+X_t^d=\varphi_1(P_t^d-X_t^d)+\varphi_2(P_t^g-X_t^d)+X_t^d\\&=(\varphi_1P_t^d+\varphi_2P_t^g)-(\varphi_1X_t^d+\varphi_2X_t^d)+X_t^d\sum_{j=1}^{m}x_{ij}^d(t+1)\\&=\varphi_1\sum_{j=1}^{m}p_{ij}^d(t)+\varphi_2\sum_{j=1}^{m}p_{ij}^g(t)-(\varphi_1\sum_{j=1}^{m}x_{ij}^d(t)+\varphi_2\sum_{j=1}^{m}x_{ij}^d(t))+\sum_{j=1}^{m}x_{ij}^d(t)\\&=\varphi_1a_i+\varphi_2a_i-(\varphi_1a_i+\varphi_2a_i)+a_i\\&=a_i\sum_{i=1}^{n}x_{ij}^d(t+1)\end{aligned}$$

$$=\varphi_1\sum_{i=1}^{n}p_{ij}^d(t)+\varphi_2\sum_{i=1}^{n}p_{ij}^g(t)-(\varphi_1\sum_{i=1}^{n}x_{ij}^d(t)+\varphi_2\sum_{i=1}^{n}x_{ij}^d(t))+\sum_{i=1}^{n}x_{ij}^d(t)$$

$$=\varphi_1 b_i+\varphi_2 b_i-(\varphi_1 b_i+\varphi_2 b_i)+b_i$$

如果 $t>0$，

$$X_{t+1}^d=V_{t+1}^d+X_t^d$$

$$=\lambda_1 V_t^d+\lambda_2[\varphi_1(P_t^d-X_t^d)+\varphi_2(P_t^g+X_t^d)]+X_t^d$$

$$=\lambda_1 X_t^d+\lambda_2[\varphi_1(P_t^d+\varphi_2 P_t^g)-(\varphi_1 X_t^d+\varphi_2 X_t^d)]+X_t^d$$

$$=\lambda_1(X_t^d-X_{t-1}^d)+\lambda_2[(\varphi_1 P_t^d+\varphi_2 P_t^g)-(\varphi_1 X_t^d+\varphi_2 X_t^d)]+X_t^d\sum_{j=1}^{m}x_{ij}^d(t+1)$$

$$=\lambda_1(\sum_{j=1}^{m}x_{ij}^d(t)-\sum_{j=1}^{m}x_{ij}^d(t-1))+\lambda_2(\varphi_1 a_i+\varphi_2 a_i-(\varphi_1 a_i+\varphi_2 a_i))+a_i$$

$$=\lambda_1(a_i-a_i)+0+a_i=a_i\sum_{j=1}^{m}x_{ij}^d(t+1)$$

$$=\lambda_1(\sum_{i=1}^{n}x_{ij}^d(t)-\sum_{i=1}^{n}x_{ij}^d(t-1))+\lambda_2(\varphi_1 b_i+\varphi_2 b_i-(\varphi_1 b_i+\varphi_2 b_i))+b_i$$

$$=\lambda_1(b_i-b_i)+0+b_i=b_i$$

因此，x_{t+1}^d在位置和速度更新方程组(11)下能够满足约束条件$\sum_{j=1}^{m}x_{ij}^d(t+1)=a_i$ 和$\sum_{i=1}^{n}x_{ij}^d(t+1)=b_j$。但是，改进后的更新原则还不能保证满足约束条件 $x_{ij}\geqslant 0, i=1,\cdots,n, j=1,\cdots,m$。为此，在下一小节中，设计了一个特别的算子来保证这个非负条件，以确保算法的适用性。

3.3 负修复算子

$$X=\begin{bmatrix} x_{11} & \cdots & x_{1i} & \cdots & x_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{k1} & \cdots & x_{ki} & \cdots & x_{km} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{l1} & \cdots & x_{li} & \cdots & x_{lm} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{ni} & \cdots & x_{nn} \end{bmatrix} \tag{12}$$

对于 PSO-NLTP 中的粒子(12)来说，如果 $x_{ki}<0, k=1,\cdots,n, i=1,\cdots,m$，可以用负修复算子来进行修正，以满足问题的实际意义。

下面是负修复时需要用到的辅助算子，用于修复粒子的指定位置。

Procedure RepairOnePos (变量 X：粒子，k,i：整数)
begin
1. 选择第 I 列中最大的元素记为 x_{lj}；
2. $x_0:=x_{ki}, x_{li}:=x_{li}-|x_0|, x_{ki}:=0$；
3. 把 k 行中的元素改为 $x_{kj}:=\begin{cases} x_{kj} & x_{kj}<|x_0| \\ x_{kj}-\dfrac{|x_{kj}|}{u} & x_{kj}\geqslant|x_0| \end{cases}$；
(u 是条件 $x_{kj}\geqslant|x_0|, j=1,\cdots,m$ 满足的次数)
4. 把 l 行中的元素改为 $x_{ij}:=\begin{cases} x_{lj} & x_{lj}<|x_0| \\ x_{lj}-\dfrac{|x_{lj}|}{u} & x_{lj}\geqslant|x_0| \end{cases}$；
(v 是条件 $x_{lj}\geqslant|x_0|, j=1,\cdots,m$ 满足的次数)
end

负修复算子算法可描述为：

Procedure RepairOneParticle (变量 X：粒子)
变量定义：i，j (整数)；
begin
如果 X 中的某些元素是负的，那么
repeat
如果 $x_{ij}<0$ 那么执行 RepairOnePos (X, i, j)；
until X 中的每个元素非负
end.

3.4 PSO 变异

变异是 GA 中的流行算子，为帮助 PSO-NLTP 中的粒子改变局部结构，以获得不同类型的解，本小节特别设计了 PSO 变异算子。PSO 变异的存在可以避免 PSO-NLTP 过快陷入局部收敛。变异算子算法设计如下：

Procedure (变量 X：粒子)
begin
1：随机得到满足条件 $0<p<n$ 和 $0<q<m$ 的 p 和 q；
2：从矩阵 X 中随机选取 p 行$\{i_1,\cdots,i_p\}$和 q 列$\{j_1,\cdots,j_q\}$形成一个小矩阵 Y ($y_{ij}, i=1,\cdots,p, j=1,\cdots,q$)；

3：$a_i^\gamma=\sum_{j\in\{j_1,\cdots,j_q\}}x_{ij}; i=i_1,\cdots,i_p$

$b_i^\gamma=\sum_{i\in\{i_1,\cdots,i_q\}}x_{ij}; j=j_1,\cdots,j_p$

4：运用初始化的过程重新分配矩阵 Y；
5：用矩阵 Y 来替换矩阵 X；
end.

4 PSO-NLTP 的仿真算例及计算结果

下面的实验中，共计算了 60 个 NLTP 算例。这些非线性算例基于文献[25]中的开放性问题和文献[24,26]，将这些文献中问题的目标函数替换成第 3 节中的非线性函数形式(7)-(10)来进行计算。文献[25]中 R. R. K. Sharma 为解决 LTP 提出了一个启发式算法。Sharma 和文献[6,27,28]中的模式对于 LTP 是有效的，而对于 NLTP，由于其非线性目标函数的复杂性，它们就失去了效力。实验均在 3.06GHz，512M DDR 内存的 PC 机上运行，并采用 Windows XP 操作系统。

表 1 PSO-NLTP，GA 和带惩罚策略的 EP 计算结果比较之一

Problem	PSO-TP Average	GA Average	EP Average	PSO-TP Time(s)	GA Time(s)	EP Time(s)
No. 1-1	8.08	8.10	8.36	0.125	0.89	0.109
No. 1-2	113.29	114.25	120.61	0.13	0.312	0.125
No. 1-4	1350.3	1350.8	1476.1	0.072	0.109	0.078
No. 1-5	205.2	206.3	216.1	0.053	0.125	0.052
No. 1-6	12.50	12.72	13.53	0.072	0.75	0.078
No. 1-7	247.2	247.6	256.9	0.098	0.32	0.093
No. 1-8	84.72	84.72	87.5	<0.001	0.015	<0.001
No. 1-9	44.65	44.65	46.2	<0.001	0.046	<0.001
No. 1-10	24.92	24.97	25.83	<0.001	0.032	<0.001
No. 2-1	155.3	155.3	168.5	<0.001	0.016	<0.001
No. 2-2	2281.5	2281.5	2696.2	<0.001	0.015	<0.001
No. 2-4	28021	28021	30020.2	<0.001	0.015	<0.001
No. 2-5	3520.2	3520.4	3583.1	<0.001	0.015	<0.001
No. 2-6	265.5	266.5	314.4	<0.001	0.015	<0.001
No. 2-7	4579.2	4584.5	5326.0	0.011	0.052	0.012
No. 2-8	432.8	432.8	432.8	<0.001	0.015	<0.001
No. 2-9	386.3	386.3	386.3	<0.001	0.031	<0.001
No. 2-10	195.3	195.3	226.0	<0.001	0.006	<0.001
No. 3-1	301.0	310.0	346.6	<0.001	0.093	0.001
No. 3-2	4650	4650	5415.2	<0.001	0.921	0.012
No. 3-4	65896.7	66123.3	68223.3	<0.001	0.105	<0.001
No. 3-5	7058.1	7066.6	7220.9	<0.001	1.015	0.001
No. 3-6	540	540	672.5	0.001	0.062	0.002
No. 3-7	9172.0	9173.2	9833.3	<0.001	0.312	<0.001
No. 3-8	1033.4	1033.4	1066.7	<0.001	0.012	<0.001
No. 3-9	933.4	933.4	1006.4	0.012	0.147	0.015
No. 3-10	480	480	480	0.016	0.046	0.004
No. 4-1	107.8	107.8	118.2	0.063	0.159	0.078
No. 4-2	1584.8	1585.2	1622	0.082	0.285	0.093
No. 4-4	19529.0	19531.3	20119	0.085	0.968	0.068
No. 4-5	2467.5	2468.2	2880.2	0.102	0.625	0.046
No. 4-6	152.0	152.1	161.9	0.163	1.046	0.167
No. 4-7	3172.5	3173.8	3227.5	0.097	0.692	0.073
No. 4-8	467.1	467.1	467.1	<0.001	0.036	<0.001
No. 4-9	376.3	376.3	382.5	<0.001	0.081	0.003
No. 4-10	205.9	205.9	227.6	0.062	0.422	0.031

实验将 M. Gen 为 TP 设计的 GA[29-31] 用于测试 NLTP 算例，以和 PSO-NLTP 进行比较，同时还与采用了基于 C. Y.

Lee 和 X. Yao 的 Lévy probability distribution 产生的变异操作的 EP[32] 进行比较。这种 EP 模式能很好地求解无约束多变量优化问题，实验中通过添加惩罚策略将它加以改进，以求解 NLTP。因为应用于不同领域，PSO-NLTP 和带惩罚策略的 EP 之间的比较只是说明 PSO-NLTP 在处理非线性 TP 的约束上是否优于 EP 的惩罚策略，而不能做出哪种算法更优的结论。这三种算法分别计算了 50 次，结果列于表 1 和表 2 中。如果找不到更好解，算法均迭代 500 次为止，这也是普遍认为算法能较好地收敛的状态。

表 1 中的算例产生于文献[25]，表 2 中的算例产生于文献[24,26]。实验中，将目标函数替换为非线性形式时，设非线性函数(9)中的 S 为 $S=\sum_{i=1}^{n}a_i/10$，设非线性函数(10)中的 S 为 $S=1$。

表 2 PSO-NLTP 和 GA 及惩罚策略 EP 的计算结果比较之二

Problem	PSO-TP Average	GA Average	EP Average	PSO-TP Time(s)	GA Time(s)	EP Time(s)
No. 1-11	1126.4	1143.09	1158.2	0.052	0.065	0.046
No. 1-12	436.8	440.3	488.3	0.192	1.312	0.203
No. 1-13	740.5	740.5	863.6	0.29	2.406	0.781
No. 1-14	2524.6	2529.0	2630.3	0.028	0.067	0.016
No. 1-15	297.6	297.9	309.2	0.066	0.178	0.058
No. 1-16	220.42	220.8	234.6	0.065	1.75	0.060
No. 2-11	50.4	51.9	64.2	<0.001	0.001	<0.001
No. 2-12	78.4	78.4	104.5	0.001	0.025	<0.001
No. 2-13	150.2	150.4	177.9	<0.001	0.015	<0.001
No. 2-14	118.2	118.2	148.4	<0.001	0.001	<0.001
No. 2-15	64.5	64.5	64.5	<0.001	0.031	<0.001
No. 2-16	47.4	47.8	53.4	<0.001	0.015	<0.001
No. 3-11	13.3	13.3	13.3	0.045	0.734	0.031
No. 3-12	21.0	21.0	26.3	0.038	0.308	0.036
No. 3-13	37.4	37.4	43.5	0.192	1.906	0.156
No. 3-14	37.8	37.8	46.7	0.018	0.578	0.008
No. 3-15	28.1	28.1	33	0.012	0.325	0.013
No. 3-16	22.8	23.0	29.6	<0.001	0.059	0.015
No. 4-11	8.8	8.8	37.4	0.002	0.106	0.001
No. 4-12	22.6	23.1	40.8	0.265	2.328	0.234
No. 4-13	51.7	52.3	72.1	0.139	2.031	0.359
No. 4-14	49.3	51.2	82.2	0.012	0.629	0.006
No. 4-15	12.06	12.06	36.58	0.024	0.484	0.026
No. 4-16	3.08	3.08	8.1	0.053	0.921	0.045

从表 1 中可以看出，三种算法在求解运输费用平均值时，PSO-NLTP 下 96%以上的平均值比 GA 和带惩罚策略的 EP 小；求解平均运算代价时，PSO-NLTP 比 GA 小得多，部分值和 EP 接近。带惩罚策略的 EP 求得的非线性 TP 解要比 PSO-NLTP 和 GA 的大，这说明 PSO-NLTP 和 GA 的算子比惩罚策略能更好地处理非线性 TP 的约束条件。同时，因为 GA 的交叉、变异算子更为复杂，带惩罚策略的 EP 比 GA 收敛得更快。PSO-NLTP 能最快获得非线性 TP 的最优解，它特别设计的算子能保证个体可行，并不断进行优化，且能快速进行全局搜索，从而使得算法对于求解非线性 TP 非常有效。

结束语 本文主要研究的是非线性运输问题(NLTP)的求解。NLTP 的解法目前并不多见。本文在经典的粒子群算法 PSO 的基础上设计了新算法 PSO-NLTP。为满足 NLTP 的供应和需求约束条件，修改了标准的粒子群位置更新和速度更新方程。为满足非负约束条件，每代进化时对 PSO 添加了一个新的负修复算子。同时，为了扩大搜索空间，还引进了变异算子。新算法可以忽略目标函数的复杂度，直接处理优化问题，因而非常适用于求解 NLTP。通过大量的非线性算例，将 PSO-NLTP 和 GA 及带惩罚策略的 EP 进行比较，PSO-NLTP 能更好地处理非线 TP 的约束条件，并能最快获得非线性 TP 的最优解，证明了 PSO-NLTP 对于求解 NLTP 非常有效。

参 考 文 献

[1] Hitchcock F L. The distribution of a product from several sources to numerous localities. J. Math. Phys. ,1941,20:224-230
[2] Kantorovich L V. Mathematical methods of organizing and planning production (in Russian). English Translation in Management Sci,1960,6:336-422
[3] 玄光南，程润伟著. 遗传算法与工程设计. 汪定伟，唐加服，黄敏，译. 科学出版社，2000：186-204
[4] Dantzig G B. Application of the simplex method to a transportation problem. // Koopmans T C, ed. Activity of production and application. NY. John Wiley & Sons,1951:359-373
[5] Orlin J B, Plotkin S A, Tardos E. Polynomial dual network simplex algorithms. Math. Program,1993,60:255-276
[6] Paparrizos K. An exterior point simplex algorithm for general linear problems. Ann. Oper. Res,1993,32:497-508
[7] Papamanthou C, Paparrizos K, Samaras N. Computational experience with exterior point algorithms for the transportation problem. Applied Mathematics and Computation,2004,158:459-475
[8] Sharma R R K, Sharma K D. A new dual based procedure for the transportation problem. European Journal of Operational Research,2000,122:611-624
[9] Vignaux G A , Michalewicz Z. A genetic algorithm for the linear transportation problem. IEEE Transactions on Systems, Man, and Cybernetics,1991,21(2):445-452
[10] Michalewicz Z, Vignaux G A, Hobbs M. A Non-standard Genetic Algorithm for the Nonlinear Transportation Problems. ORSA Journal on Computing,1991,3(4):307-316
[11] Li Y Z, Ida K C, Gen M. Improved genetic algorithm for solving multi objective solid transportation problem with fuzzy numbers. Computers ind. Engng,1997,33(3/4):589-592
[12] Gen M, Ida K, Kono E, et al. Solving bicriteria solid transportation problem by genetic algorithms // Proceedings of the 16th International Conference on computers and industrial engineering. Ashikaga, Japan,1994:572-575
[13] Yang X, Gen M. Evolution program for bicriteria transportation problem // Proceedings of the 16th International Conference on computers and industrial engineering. Ashikaga, Japan,1994:451-454
[14] Tsuchiya K, Nishyama T, Tsujita K. A deterministic annealing algorithm for a Combinatorial optimization problem using replicator equations [J]. Physica D,2001,149:161-173
[15] Liu B. Dependent chance programming-a class of Stochastic programming [J]. Computers&Mathematics with Applications, 1997,34(12):89-104
[16] Backler G G. Mode split in the anglo European unitized freight market [J]. Transportation Planning Methods,1993,9:271-282
[17] Master I, Sviden O, Wegener M. Transport planning for equity and sustainability [J]. Transportation Planning and Technology,1993,22(17):319-330
[18] 帅斌. 基于系统利润最优的运价规则安排的研究[J]. 西南交通大学学报，2001(2)：92-95
[19] 何德权. 运输价格理论及其定价模型研究[D]. 成都：西南交通大学，2000
[20] 杨菊花，盖宇仙. 用最小费用最大流理论确定铁路货物运价问题的研究. 兰州交通大学学报，2005,24(1):139-142
[21] Kennedy J, Eberhart R C. Particle Swarm Optimization[C] // Proceedings of IEEE International Conference on Neutral Networks. Perth, Australia,1995:1942-1948
[22] Kennedy J. The Particle Swarm. Social Adaptation of Knowledge [C] // Proceedings of IEEE International Conference on Evolutionary Computation . Indianapolis, Indiana,1997
[23] 玄光南，程润伟著. 遗传算法与工程优化. 于歆杰，周根贵，译. 清华大学出版社，2004：226-254
[24] 魏国华. 实用运筹学. 上海：复旦大学出版社，1987：98-120
[25] Sharma R R K, Sharma K D. A new dual based procedure for the transportation problem. European Journal of Operational Research,2000,122:611-624
[26] 郭强. 运输问题的一种新的迭代算法. 计算机工程与应用，2004：57-58
[27] Orlin J B, Plotkin S A, Tardos E. Polynomial dual network simplex algorithms, Math, Program,1993,60:255-276
[28] Papamanthou , Paparrizos C, Samaras C. Computational experience with exterior point algorithms for the transportation problem. Applied Mathematics and Computation,2004,158:459-475
[29] Vignaux G A, Michalewicz Z. A genetic algorithm for the linear transportation problem. IEEE Transactions on Systems, Man, Cybernetics,1991,21(2):445-452
[30] Michalewicz Z, Vignaux G A, Hobbs M. A Non-standard Genetic Algorithm for the Nonlinear Transportation Problems. ORSA Journal on Computing,1991,3 (4):307-316
[31] Li Y Z, Ida K C, Gen M. Improved genetic algorithm for solving multi objective solid transportation problem with fuzzy numbers. Computers Industrial Engineering,1997,33 (3/4):589-592
[32] Lee C Y, Yao X. Evolutionary Programming Using Mutations Based on the Lévy Probability Distribution. IEEE Transactions on Evolutionary Computation,2004,8(1)