# 基于 PLSR-IBPSO 的有约束组合优化问题求解方法研究——以人机交互界面设计为例*

郭　伏，屈庆星
（东北大学 工商管理学院，辽宁 沈阳 110167）

**摘　要**　本文针对工程与管理领域中的有约束组合优化问题，考虑变量维数大且变量间存在多重相关性的因素，以及变量间有强约束关系的特点，建立新的符合实际问题的数学模型。运用偏最小二乘回归（partial least squares regression，PLSR）构建变量之间的回归模型，提出利用二进制粒子群算法优化变量 $m$ 的 $n$ 个水平，以期达到目标函数最优的要求，并依据求解问题的特殊性对该算法进行改进。制定新的初始种群产生策略，保证在可行解空间内进行寻优；引入动态惯性权重，保障算法具有更快的收敛性能；修改种群更新机制并加入判别函数，确保种群每次更新后都满足模型中的等式约束。通过算例和数值仿真分析，证实该算法是有效的，并能够得到较好的结果。

**关键词**　组合优化，PLSR，二进制粒子群算法，约束优化，产品设计优化
**中图分类号**　O224；TB472

## 1　引言

组合优化问题在管理实践中有着广泛的应用，同时是管理科学中的重要研究问题。组合优化是一种离散最优化问题，在规划、调度、资源分配、决策等问题中有着非常广泛的应用[1]。典型的组合优化问题有旅行商问题（traveling salesman problem，TSP）、背包问题（knapsack problem，KP）、装箱问题（bin packing problem，BPP）、聚类问题（clustering problem，CP）等。有约束组合优化问题（constrained combinatorial optimization problem，CCOP）是指在离散的、有限的数学结构上，怎样寻找一个（或一组）满足给定约束条件并使其目标函数值达到最优解的问题[2]。该问题广泛存在于工程与管理的各个领域，如产品设计优化、系统故障检测、超大规模集成电路和航班机组排班等，所以对该问题进行深入研究，具有重要的理论意义和应用价值。

文献研究表明，目前求解有约束组合优化问题的方法可分为精确算法和启发式算法。对于变量维数较小的有约束组合优化问题，传统的求解方法有隐枚举法[3]、分支定界法[4, 5]和割平面法[6-8]等，可以求得问题的精确解；对于变量维数较大的有约束组合优化问题，已被证明属于极强的 NP-Hard 问题[9]，传统方法将难以求解。已有学者应用遗传算法（genetic algorithm，GA）[10, 11]、蚁群优化（ant colony optimization，ACO）[12, 13]、模拟退火（simulated annealing，SA）[14, 15]、禁忌搜索（tabu search，TS）[16, 17]和粒子群优化（particle swarm optimization，PSO）[18, 19]等启发式算法求解规模较大的有约束组合优化问题，启发式算法虽不能保证求得问题的最优解，但因其求解速度快，可求得问题的近似最优解，被认为是解决

* 基金项目：国家自然科学基金（71771045，71471033）。
通信作者：屈庆星，东北大学工商管理学院，博士研究生，E-mail：yantaiquqingxing@163.com。

较大规模有约束组合优化问题的有效方法。

粒子群算法作为一种典型的启发式算法，已在各种组合优化问题中得到了广泛的应用，该算法最初由 Kennedy 和 Eberhart 提出[20, 21]，后来学者们针对该算法性能的局限性进行了改进和完善，以增加算法的精确度，加强算法的局部搜索能力和加快算法的搜索速度。Angeline 提出了一种基于锦标赛算子的选择机制，加快了粒子群算法的更新过程[22]；Angeline 将 GA 中交叉算子的思想引入粒子群算法中，避免了该算法的早熟现象，但算法的收敛速度较慢[23]；Kennedy 和 Mendes 提出了星形结构、环形结构和金字塔结构等不同的拓扑结构，用以改善粒子群算法的收敛速度[24]；Suganthan 提出了一种基于动态邻域的粒子群算法，进一步提高了算法的性能，避免了算法过早收敛的现象[25]。

虽然启发式算法能够较好地解决较大规模的有约束组合优化问题，但大多数仍采用实数域方法进行取整运算[26]。这种方法在一些测试实例中取得了较好的结果，但是在另外一些测试实例中很难取得好的结果[27]。对于离散变量的组合优化问题，Kennedy 和 Eberhart 在粒子群算法的基础上发展了二进制粒子群算法（binary particle swarm optimization，BPSO）来解决这一类问题[28]。虽然 BPSO 可以保证求得问题的最优解为整数解，但其产生的初始种群很难保证在可行解空间内，该算法容易掉入局部最优解陷阱，并且可行解无法确保满足约束条件。

本文针对 BPSO 在求解有约束组合优化问题时的局限性，通过制定新的初始种群产生策略，引入动态惯性权重，修改种群更新机制并加入判别函数，提出一种在整数可行解中直接进行进化计算的改进二进制粒子群算法（improved binary particle swarm optimization，IBPSO），该方法保证了粒子群在进化过程中始终被控制在整数可行解空间内，避免了不必要的实数域搜索，加快了收敛速度。本文通过算例和数值仿真分析，证实该算法是有效的，并能够得到较好的结果。

## 2 问题描述与数学模型

有约束组合优化问题求解目标是从有限个或可数无限个离散解的集合中求出最优解。一般表示为：令 $X=\{x_1,x_2,\cdots,x_D\}$ 为 $D$ 维的可行解空间，$f(S)$ 表示自变量处于 $S$ 状态时对应的目标函数值，目标是寻找最优解 $S^*$，满足 $f(S^*)=\max f(S)$ 或 $f(S^*)=\min f(S)$，其中 $S^*\in X$，本文以 $f(S^*)=\max f(S)$ 为最优解。

一般来说，有约束组合优化问题往往伴有大量的局部极值点，并且是不可微的、不连续的、多维的和高度非线性的 NP-Hard 问题。从理论上来讲，如果 $X=\{x_1,x_2,\cdots,x_D\}$ 是有限集合的话，只要在可行解空间内遍历所有的组合，就能找到最优解。然而，随着问题规模的扩大，$X$ 中可行解的个数会呈指数增长，很难在多项式时间内获得问题的最优解。本文以 0-1 有约束组合优化问题为例，研究在约束条件下求解组合优化问题最优解的有效算法，构建的目标函数为

$$\max f(X)=\max\left(\sum_{n=1}^{N_1}x_{1n}\tau_1+\sum_{n=2}^{N_2}x_{2n}\tau_2+\cdots+\sum_{n=1}^{N_M}x_{Mn}\tau_M\right) \tag{1}$$

约束条件：

$$\sum_{n=1}^{N}x_{mn}=1,\quad x_{mn}=0\text{ 或 }1 \tag{2}$$

其中，$m=1,2,\cdots,M$ 表示第 $m$ 个项目；$n=1,2,\cdots,N$ 表示第 $m$ 个项目的第 $n$ 个水平；$\tau_m$ 表示第 $m$ 个项目 $n$ 个水平的系数向量。同时令

$$x_{mn}=\begin{cases}1, & \text{项目}m\text{的水平}n\text{被选入}\\ 0, & \text{其他}\end{cases}$$

满足约束条件的可行解 $x_{mn}$ 可写成列向量形式，即解向量，上述目标函数的一个可行解向量如下：

$$x_{mn}=[1010100101010101001010101001010101 00]^{\mathrm{T}}$$

# 3 目标函数构建

在工程与管理领域，常会遇到一些自变量个数多、可获取变量观察值样本数量少，并且自变量之间存在多重相关性的回归建模问题。考虑变量维数大且变量间存在多重相关性的因素，以及变量间有强约束关系的特点，运用 PLSR 构建有约束组合优化问题的目标函数，可以有效地解决自变量之间的多重相关性问题，适合在样本个数小于自变量个数的情况下进行回归建模分析。当自变量个数多、可获取变量观察值样本个数少时，尤其当自变量个数大于样本个数时，其他统计分析方法此时已无法适用，而该方法仍然相当有效。

## 3.1 PLSR 的基本思想

PLSR 的基本思想如下。假设有 $q$ 个因变量 $y_1,y_2,\cdots,y_q$，$p$ 个自变量 $x_1,x_2,\cdots,x_p$。首先，分别从自变量集合中提取成分 $t_1$，从因变量集合中提取成分 $\mu_1$，并且要求 $t_1$ 和 $\mu_1$ 的相关性达到最大。其次，建立因变量 $y_1,y_2,\cdots,y_q$ 与 $t_1$ 的回归方程，如果回归方程已经达到满意的结果，则算法终止，否则将进行下一个成分的提取。最后，若从自变量集合最终提取 $r$ 个成分 $t_1,t_2,\cdots,t_r$，通过建立 $y_1,y_2,\cdots,y_q$ 与 $t_1,t_2,\cdots,t_r$ 的回归方程，还原成 $y_1,y_2,\cdots,y_q$ 与 $x_1,x_2,\cdots,x_p$ 的回归方程。

## 3.2 PLSR 构建目标函数

PLSR 构建目标函数的具体步骤如下。

步骤 1：数据预处理。

不妨设，$p$ 个自变量 $x_1,x_2,\cdots,x_p$ 与 $q$ 个因变量 $y_1,y_2,\cdots,y_q$ 的标准化矩阵分别为 $E_0$ 与 $F_0$。

步骤 2：提取成分 $t_1$ 和 $\mu_1$。

$t_1$ 是自变量集 $X=(x_1,x_2,\cdots,x_p)^{\mathrm{T}}$ 的线性组合，$t_1=a_{11}x_1+a_{12}x_2+\cdots+a_{1p}x_p=a_1^{\mathrm{T}}X$；$\mu_1$ 是因变量 $Y=(y_1,y_2,\cdots,y_q)^{\mathrm{T}}$ 的线性组合，$\mu_1=b_{11}y_1+b_{12}y_2+\cdots+b_{1q}y_q=v_1^{\mathrm{T}}Y$。要求 $t_1$ 和 $\mu_1$ 尽可能多地提取变量信息，$t_1$ 和 $\mu_1$ 的相关性达到最大。令 $\hat{t}_1$ 和 $\hat{\mu}_1$ 表示成分 $t_1$ 和 $\mu_1$ 的得分向量，即

$$\hat{t}_1=E_0a_1=\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1p}\\ x_{21} & x_{22} & \cdots & x_{2p}\\ \vdots & \vdots & & \vdots\\ x_{i1} & x_{i2} & \cdots & x_{ip}\end{bmatrix}\begin{bmatrix}a_{11}\\ a_{12}\\ \vdots\\ a_{1p}\end{bmatrix}=\begin{bmatrix}\hat{t}_{11}\\ \hat{t}_{21}\\ \vdots\\ \hat{t}_{i1}\end{bmatrix} \tag{3}$$

$$\hat{\mu}_1=F_0b_1=\begin{bmatrix}y_{11} & y_{12} & \cdots & y_{1q}\\ y_{21} & y_{22} & \cdots & y_{2q}\\ \vdots & \vdots & & \vdots\\ y_{i1} & y_{i2} & \cdots & y_{iq}\end{bmatrix}\begin{bmatrix}b_{11}\\ b_{12}\\ \vdots\\ b_{1q}\end{bmatrix}=\begin{bmatrix}\hat{\mu}_{11}\\ \hat{\mu}_{21}\\ \vdots\\ \hat{\mu}_{i1}\end{bmatrix} \tag{4}$$

此外，成分 $t_1$ 和 $\mu_1$ 的协方差 $\mathrm{Cov}(t_1,\mu_1)$ 可用 $\hat{t}_1$ 和 $\hat{\mu}_1$ 的内积来计算，因此，满足提取成分 $t_1$ 和 $\mu_1$ 的两

个要求等价于以下条件极值问题：

$$\max \mathrm{Cov}(t_1,\mu_1)=\max <\hat{t}_1,\hat{\mu}_1>=\max <E_\circ a_1,F_\circ b_1>=\max(a_1^{\mathrm{T}}E_\circ^{\mathrm{T}}F_\circ b_1) \tag{5}$$

$$a_1^{\mathrm{T}}a_1=\|a_1\|^2=1,\quad b_1^{\mathrm{T}}b_1=\|b_1\|^2=1 \tag{6}$$

性质 1　$\exists\theta=E_\circ^{\mathrm{T}}F_\circ F_\circ^{\mathrm{T}}E_\circ$，使 $\theta^2$ 为矩阵 $E_\circ^{\mathrm{T}}F_\circ F_\circ^{\mathrm{T}}E_\circ$ 的最大特征值，$a_1$ 为该特征值对应的单位特征向量；同理，$b_1$ 为对应于矩阵 $E_\circ^{\mathrm{T}}F_\circ F_\circ^{\mathrm{T}}E_\circ$ 最大特征值 $\theta^2$ 的单位特征向量。

步骤 3：建立回归方程

$$E_\circ=\hat{t}_1\alpha_1^{\mathrm{T}}+E_1 \tag{7}$$

$$F_\circ=\hat{\mu}_1\beta_1^{\mathrm{T}}+F_1 \tag{8}$$

其中，$\alpha_1=(\alpha_{11},\alpha_{12},\cdots,\alpha_{1p})^{\mathrm{T}}$，$\beta_1=(\beta_{11},\beta_{12},\cdots,\beta_{1q})^{\mathrm{T}}$ 表示回归方程中的系数向量；$E_1$ 和 $F_1$ 表示残差矩阵。$\alpha_1$，$\beta_1$ 的最小二乘估计为

$$\alpha_1=\frac{\hat{t}_1}{\|\hat{t}_1\|^2}E_\circ^{\mathrm{T}},\quad \beta_1=\frac{\hat{\mu}_1}{\|\hat{\mu}_1\|^2}F_\circ^{\mathrm{T}} \tag{9}$$

步骤 4：用残差矩阵 $E_1$ 和 $F_1$ 分别代替 $E_\circ$ 和 $F_\circ$ 重复步骤 2 与步骤 3。

记 $\hat{E}_\circ=\hat{t}_1\alpha_1^{\mathrm{T}}$，$\hat{F}_\circ=\hat{\mu}_1\beta_1^{\mathrm{T}}$，则残差矩阵 $E_1=E_\circ-\hat{E}_\circ$，$F_1=F_\circ-\hat{F}_\circ$。如果残差矩阵 $E_1$ 和 $F_1$ 中元素的绝对值近似为 0，则认为用成分 $t_1$ 和 $\mu_1$ 建立的回归方程已经达到满意的结果，可以停止抽取成分。否则用残差矩阵 $E_1$ 和 $F_1$ 代替 $E_\circ$ 和 $F_\circ$ 重复步骤 2 与步骤 3 继续提取新成分，建立回归方程。

步骤 5：建立 PLSR 方程。

不妨设分别从 $E_\circ$ 和 $F_\circ$ 中提取了 $r$ 个成分 $t_1,t_2,\cdots,t_r$ 和 $\mu_1,\mu_2,\cdots,\mu_r$，使

$$E_\circ=\hat{t}_1\alpha_1^{\mathrm{T}}+\hat{t}_2\alpha_2^{\mathrm{T}}+\cdots+\hat{t}_r\alpha_r^{\mathrm{T}}+E_r \tag{10}$$

$$F_\circ=\hat{\mu}_1\beta_1^{\mathrm{T}}+\hat{\mu}_2\beta_2^{\mathrm{T}}+\cdots+\hat{\mu}_r\beta_r^{\mathrm{T}}+F_r \tag{11}$$

把 $t_k=a_{k1}x_1+a_{k2}x_2+\cdots+a_{kp}x_p$（$k=1,2,\cdots,r$）代入 $Y=t_1\beta_1+t_2\beta_2+\cdots+t_r\beta_r$，可以得到 $q$ 个因变量的 PLSR 方程式。

$$y_j=a_{j1}x_1+a_{j2}x_2+\cdots+a_{jp}x_p,\quad j=1,2,\cdots,q \tag{12}$$

## 3.3 交叉有效性检验

交叉有效性检验（cross validation，CV）用来确定 PLSR 方程中所应提取的成分个数。定义含 $h$ 个成分的 PLSR 方程的交叉有效性为 $Q_h^2=1-\dfrac{\mathrm{PRESS}(h)}{\mathrm{SS}(h-1)}$，当 $Q_h^2\geqslant 0.0975$，表示成分 $t_h$ 的边际贡献显著，应提取第（$h+1$）个成分，否则应停止提取成分。其中，$\mathrm{PRESS}(h)=\sum_{j=1}^{q}\mathrm{PRESS}_j(h)$，表示舍去第 $\eta$ 个观测值（$\eta=1,2,\cdots,i$）后，用余下的（$i-1$）个观测值构建 PLSR 方程并提取 $h$ 个成分时，因变量 $Y=(y_1,y_2,\cdots,y_q)^{\mathrm{T}}$ 的预测残差平方和（prediction residual error sum of squares，PRESS）；$\mathrm{SS}(h)=\sum_{j=1}^{q}\mathrm{SS}_j(h)$，表示采用所有样本观测值构建 PLSR 方程并提取 $h$ 个成分时，因变量 $Y$ 的预测残差平方和。

# 4 模型求解

目前求解有约束组合优化问题的方法可分为精确算法和启发式算法。对于变量维数较小的有约束组合优化问题，传统的求解方法可以求得问题的精确解；对于变量维数较大的有约束组合优化问题，传统方法将难以求解，并且时间成本较大。启发式算法虽不一定求得问题的最优解，但因其求解速度快，可求得问题的近似最优解，被认为是解决较大规模有约束组合优化问题的有效方法。本文在 BPSO 的基础上，通过制定新的初始种群产生策略，引入动态惯性权重，修改种群更新机制并加入判别函数，提出了一种在整数可行解中直接进行进化计算的 IBPSO。

## 4.1 粒子群算法

粒子群算法源于对鸟群捕食行为的研究，找到食物最简单有效的方法是搜寻当前距离食物最近的每一只鸟的周围区域[20, 29]。粒子群算法中每个粒子位置代表了求解问题的一个潜在可行解，并用位置、速度和适应度值表示该粒子的特征。粒子的速度决定了粒子的方向和位置，每个粒子对应一个由适应度函数决定的粒子适应度值。粒子每更新一次位置，就重新计算粒子适应度值，并且通过比较新粒子的适应度值更新个体极值 $p_{\text{best}}$ 和群体极值 $g_{\text{best}}$ 的位置。

不妨设 $D$ 维空间中，由 $n$ 个粒子组成的种群 $X=(x_1,x_2,\cdots,x_n)$，其中第 $i$ 个粒子表示一个 $D$ 维的向量 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})^{\mathrm{T}}$，代表第 $i$ 个粒子位置，即求解问题的一个潜在可行解。由目标函数可求得粒子每个位置 $X_i$ 对应的适应度值。第 $i$ 个粒子的速度为 $V_i=(v_{i1},v_{i2},\cdots,v_{iD})^{\mathrm{T}}$，其个体极值为 $P_i=(P_{i1},P_{i2},\cdots,P_{iD})^{\mathrm{T}}$，种群的群体极值为 $P_g=(P_{g1},P_{g2},\cdots,P_{gD})^{\mathrm{T}}$。粒子每次更新时，通过式（13）和式（14）更新自身的速度与位置。

$$v_{id}^{k+1}=v_{id}^{k}+c_1r_1(P_{id}^{k}-x_{id}^{k})+c_2r_2(P_{gd}^{k}-x_{id}^{k}) \tag{13}$$

$$x_{id}^{k+1}=x_{id}^{k}+v_{id}^{k+1} \tag{14}$$

其中，$d=1,2,\cdots,D$；$i=1,2,\cdots,n$；$k$ 表示当前迭代次数；$v_{id}$ 表示粒子的速度；$c_1$ 和 $c_2$ 是非负的常数，表示学习因子；$r_1$ 和 $r_2$ 表示分布于 [0,1] 区间的随机数。

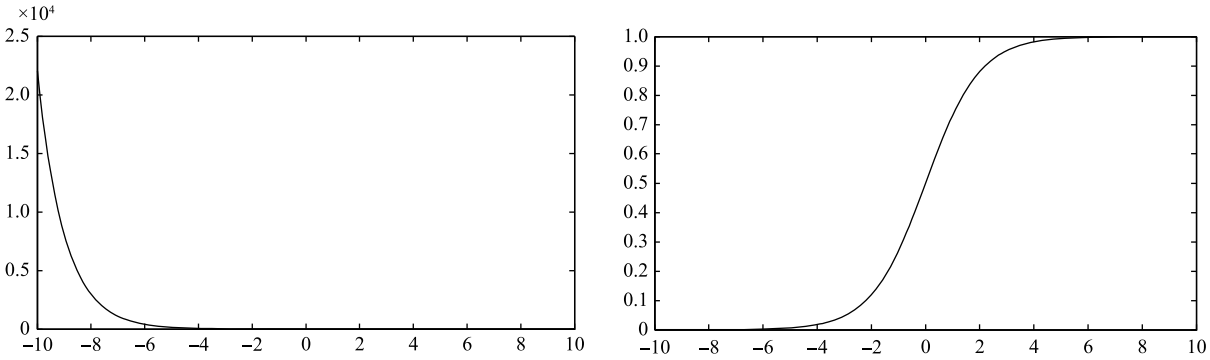
## 4.2 BPSO

粒子群算法是用来对连续函数进行优化的，而工程与管理领域的许多实际问题是离散变量的组合优化问题。Kennedy 和 Eberhart 在粒子群算法的基础上发展了 BPSO 来解决这一类问题[28]。与粒子群算法不同的是，BPSO 使用了一种具有概率性质的位置更新方程，而速度更新与粒子群算法类似。式（14）修改为

$$x_{id}^{k+1}=\begin{cases}1, & \text{rand}<\text{Sig}(v_{id}^{k+1})\\ 0, & \text{rand}\geqslant\text{Sig}(v_{id}^{k+1})\end{cases} \tag{15}$$

其中，rand 表示区间（0，1）的随机数。在 BPSO 中，粒子位置 $x_{id}^{k}$、局部极值 $p_{id}^{k}$、全局极值 $p_{gd}^{k}$ 的取值为离散的 0 或 1，而 $v_{id}^{k}$ 无需离散化，$v_{id}^{k}$ 的大小决定粒子位置取值为 0 或 1 的概率，速度越大则粒子位置取值为 1 的概率越大，反之越小。为了体现粒子速度与概率的这种关系，BPSO 采用 Sigmoid 函数作为传递函数，Sigmoid 函数的定义如式（16）所示，其函数图像如图 1 所示。

$$\text{Sig}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{16}$$




（a） $f(x)=\mathrm{e}^{-x}$ （b） $\mathrm{Sig}(x)=\frac{1}{1+\mathrm{e}^{-x}}$

图 1 Sigmoid 函数图像

由图 1（b）Sigmoid 函数图像可以看出，为防止 Sigmoid 函数饱和，可将 $v_{id}^{k}$ 限制在区间[–4,4]之内。本文将 $v_{\max}$ 设定为 4，$v_{\min}$ 设定为−4，这样对应的 Sigmoid 函数式（16）修改为

$$\mathrm{Sig}(v_{id}^{k+1})=\begin{cases}0.982, & v_{id}^{k+1}>4\\ \dfrac{1}{1+\mathrm{e}^{-v_{id}^{k+1}}}, & -4\leqslant v_{id}^{k+1}\leqslant 4\\ 0.018, & v_{id}^{k+1}<4\end{cases} \tag{17}$$

## 4.3 IBPSO

虽然 BPSO 可以保证求得问题的最优解为整数解，但其产生的初始种群很难保证在可行解空间内，该算法容易掉入局部最优解陷阱，并且可行解无法确保满足约束条件。针对 0-1 有约束组合优化问题，本文采用 IBPSO 进行求解。本文在 BPSO 的基础上，制定新的初始种群产生策略，保证在可行解空间内进行寻优；引入动态惯性权重，保障算法具有更快的收敛性能；修改种群更新机制并加入判别函数，确保种群每次更新后都满足模型中的等式约束。

1）制定新的初始种群产生策略

本文采用新的策略来生成初始种群，以保证产生的初始种群为目标函数的可行解，使算法在可行解空间内开始寻优。考虑约束条件式（2）为强约束关系，项目 $m$ 有 $n$ 个水平，并且有且只有一个水平为 1，该项目的其他水平均为 0，项目 $m$ 的 $n$ 个水平取值为 1 的概率均为 $\frac{1}{n}$。不妨设 $[1,0,\cdots,0]$ 为项目 $m$ 的一个可行解，由于项目 $m$ 的 $n$ 个水平取值为 1 是随机的，且满足 $\sum_{n=1}^{N}x_{mn}=1$，$m=1,2,\cdots,M$，令 $x_{mn}=[1,0,\cdots,0]$，定义函数 $R(x_{mn})$ 使向量 $x_{mn}$ 中元素随机重排，求解 $x'_{mn}=R(x_{mn})$，则向量 $G_{\circ}=[x'_{1n_1},x'_{2n_2},\cdots,x'_{Mn_M}]$ 为可行的初始种群。

2）引入动态惯性权重

惯性权重 $\omega^{k}$ 体现的是粒子继承先前速度的能力，Shi 和 Eberhart 最先将惯性权重 $\omega^{k}$ 引入粒子群算法中，指出较大的惯性权重值有利于全局搜索，而较小的惯性权重值则更利于局部搜索，并且提出了线性递减惯性权重（linear decreasing inertia weight，LDIW），如式（18）所示，其可以更好地平衡算法的全局搜索能力与局部搜索能力[21]。

$$\omega^k = \omega_s - (\omega_s - \omega_e)\left(\frac{k}{T_{\max}}\right) \tag{18}$$

其中，$\omega_s$ 表示初始惯性权重；$\omega_e$ 表示迭代至最大次数时的惯性权重；$k$ 表示当前迭代次数；$T_{\max}$ 表示最大迭代次数。但是，线性递减惯性权重只是一种经验做法，较容易使粒子群算法掉入局部最优解陷阱。如图 2 所示，当粒子 $x_{id}$ 由 $k$ 迭代到（$k+1$）时，若该粒子的适应度值由 $F(x_{id}^k)$ 进化到 $F_1(x_{id}^{k+1})$，较小的惯性权重值可以使算法更精确地进行局部搜索以取得最优解 $F^*(x_{id})$；反之，若该粒子的适应度值由 $F(x_{id}^k)$ 进化到 $F_2(x_{id}^{k+1})$，此时如果仍使用较小的惯性权重值，算法将容易陷入局部最优解。

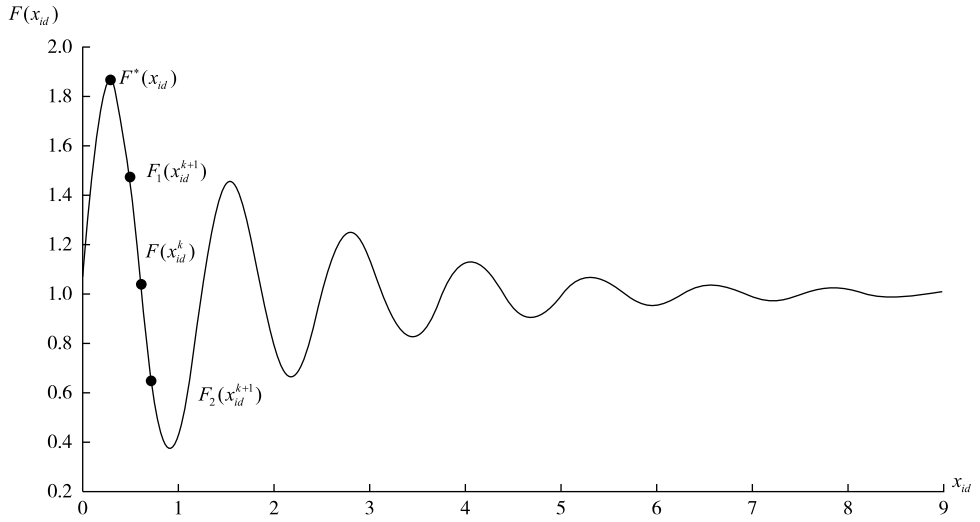



图 2　局部最优解陷阱

为实现算法全局搜索能力和局部搜索能力之间的有效平衡，避免粒子群算法过早陷入局部最优解陷阱，本文引入动态惯性权重，保障算法具有更快的收敛性能，如式（19）所示。

$$\omega_{k+1} = \begin{cases} \omega_e\left(\dfrac{\omega_s}{\omega_e}\right)^{1/(1+10(k+1)/G_{\max})}, & F(x_{id}^{k+1}) \succ F(x_{id}^k) \\ \omega_s - (\omega_s - \omega_e)\left(\dfrac{k}{G_{\max}}\right)^2, & F(x_{id}^{k+1}) \prec F(x_{id}^k) \end{cases} \tag{19}$$

其中，定义运算符号“A≻B”或“B≺A”表示“A 优于 B”或“B 不优于 A”；$F(x_{id})$ 表示适应度函数。则式（20）代替粒子群速度更新式（14）。

$$v_{id}^{k+1} = \omega_{k+1} v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \tag{20}$$

3）修改种群更新机制

为确保满足约束条件的初始种群在进行速度更新和位置更新之后，产生的新种群仍能够满足约束条件，需要对 BPSO 的种群更新机制进行改进，在粒子位置更新式（15）中加入判别函数 $\Phi$，修改为

$$x_{id}^{k+1} = \begin{cases} 1, & \text{rand} < \text{Sig}(v_{id}^{k+1}) \\ 0, & \text{rand} \geqslant \text{Sig}(v_{id}^{k+1}) \end{cases} \cap \Phi(x_{id}^{k+1}) \tag{21}$$

其中，有约束条件下 0-1 组合优化问题，判别函数定义为

$$\Phi(x_{mn}^{k+1})=\begin{cases}1, & \sum_{n=1}^{N-1}x_{mn}^{k+1}=0,\ x_{mn}^{k+1}\in\{0,1\}\\ 0, & \sum_{n=1}^{N-1}x_{mn}^{k+1}=1,\ x_{mn}^{k+1}\in\{0,1\}\end{cases} \tag{22}$$

其中，$x_{mn}^{k+1}$ 表示第（$k+1$）次迭代之后，项目 $m$ 第 $n$ 个水平的取值。

### 4.4 算法步骤

求解变量维数较大的有约束条件下 0-1 组合优化问题是一个 NP-Hard 问题，本文利用 IBPSO 求解含有 PLSR 方程构建目标函数的有约束组合优化问题的最优解。其中，适应度函数为目标函数式（1）中的 $f(X)=\sum_{n=1}^{N_1}x_{1n}\tau_1+\sum_{n=2}^{N_2}x_{2n}\tau_2+\cdots+\sum_{n=1}^{N_M}x_{Mn}\tau_M$。适应度函数值越大，说明 $f(X)$ 越接近问题的最优解。算法步骤如表 1 所示。

**表 1 IBPSO 步骤设计**

| | |
|---|---|
| 步骤 1 | 参数初始化<br>设置种群规模 $G_{\max}$；最大迭代次数 $T_{\max}$；初始惯性权重 $\omega_s$，终止惯性权重 $\omega_e$，学习因子 $c_1$，$c_2$ |
| 步骤 2 | 种群粒子初始化<br>步骤 2.1 定义随机元素重排函数 $R(x_{mn})$，使 $x'_{mn}=R(x_{mn})$，其中 $x_{mn}=[1,0,\cdots,0]$，$m=1,2,\cdots,M$，则初始有效种群为 $G_\circ=[x'_{1n_1},x'_{2n_2},\cdots,x'_{Mn_M}]$<br>步骤 2.2 随机生成粒子的初始速度 $v_{mn}$，满足 $-v_{\max}\leqslant v_{mn}\leqslant v_{\max}$ |
| 步骤 3 | 粒子适应度值计算<br>$p_{\text{best}}$ 表示个体最优值，$g_{\text{best}}$ 表示种群最优值 |
| 步骤 4 | 利用式（15）~式（17）调整粒子位置 |
| 步骤 5 | 利用式（21）~式（22）判断粒子 $x_{mn}^{k+1}$ 是否在可行解内，若否，则返回步骤 2.1 和步骤 2.2，重新初始化粒子；若是，则进入步骤 6 |
| 步骤 6 | 计算新粒子适应度值<br>步骤 6.1 更新粒子适应度值<br>若 $F(x_{mn}^{k+1})\succ F(x_{mn}^{k})$，则 $F^*(x_{mn})=F(x_{mn}^{k+1})$，$x_{mn}^*=x_{mn}$<br>步骤 6.2 引入动态惯性权重 $\omega_{k+1}$<br>当 $F(x_{mn}^{k+1})\succ F(x_{mn}^{k})$ 时，用公式 $\omega_{k+1}=\omega_e\left(\frac{\omega_s}{\omega_e}\right)^{1/(1+10(k+1)/G_{\max})}$ 更新粒子速度；反之，使用公式 $\omega_{k+1}=\omega_s-(\omega_s-\omega_e)\left(\frac{k}{G_{\max}}\right)^2$ 计算。转入步骤 4 |
| 步骤 7 | 当 $k>T_{\max}$ 时，停止运算，输出 $F^*(x_{mn})$，$x_{mn}^*$ |

## 5 算例分析

随着消费水平的提高，用户在购买或使用产品的过程中越来越注重产品带来的情感体验，特别是网站、应用软件等人机交互界面，其设计水平直接影响用户的情感体验，进而影响用户偏好。算例分析选择服装类电子商务网站设计优化问题，通过本文提出的建模及优化方法对其进行优化设计。

### 5.1 变量选取及数据收集

邀请网站设计人员、人因工程领域专家组成焦点小组，运用形态分析法选取影响用户情感体验的网页界面设计变量及其水平，如表 2 所示。

**表 2　影响用户情感体验的网页界面设计变量及其水平**

| 区域位置 | 网页界面设计变量 | 变量水平 |
|---|---|---|
| 商品搜索栏 | 网站 logo 位置（ $x_1$ ） | 左侧（ $x_{11}$ ） |
| | | 中部（ $x_{12}$ ） |
| | 导航背景颜色（ $x_2$ ） | 与网页背景颜色不同色系（橘红色）（ $x_{21}$ ） |
| | | 与网页背景颜色同色系（灰色）（ $x_{22}$ ） |
| 商品信息栏 | 小图辅助展示位置（ $x_3$ ） | 大图下部（ $x_{31}$ ） |
| | | 大图左侧（ $x_{32}$ ） |
| | | 大图右侧（ $x_{33}$ ） |
| | 商品名称位置（ $x_4$ ） | 大图上部（ $x_{41}$ ） |
| | | 商品文字信息上部（ $x_{42}$ ） |
| | 商品名称文字大小相对于其他信息文字（ $x_5$ ） | 大且很明显（ $x_{51}$ ） |
| | | 较大且不明显（ $x_{52}$ ） |
| | 价格呈现形式（ $x_6$ ） | 背景突出，字体加大、变粗、变色（ $x_{61}$ ） |
| | | 字体加大、变粗、变色（ $x_{62}$ ） |
| | 商品原价提醒（ $x_7$ ） | 有（ $x_{71}$ ） |
| | | 无（ $x_{72}$ ） |
| | 选择颜色时的颜色示例（ $x_8$ ） | 商品附颜色示例且标示颜色名称（ $x_{81}$ ） |
| | | 商品附颜色示例且鼠标悬浮显示颜色名称（ $x_{82}$ ） |
| | | 商品附颜色示例且鼠标悬浮无效果（ $x_{83}$ ） |
| 相关信息栏 | 相关信息标签背景颜色（ $x_9$ ） | 与网页背景颜色不同色系（橘红色）（ $x_{91}$ ） |
| | | 与网页背景颜色同色系（灰色）（ $x_{92}$ ） |
| | 相关信息上部的店铺同类商品信息（ $x_{10}$ ） | 有（ $x_{101}$ ） |
| | | 无（ $x_{102}$ ） |
| | 试穿体验或模特档案（ $x_{11}$ ） | 有（ $x_{111}$ ） |
| | | 无（ $x_{112}$ ） |
| | 模特展示和平铺展示屏数（ $x_{12}$ ） | 4 屏以下（ $x_{121}$ ） |
| | | 4~7 屏（ $x_{122}$ ） |
| | | 7 屏以上（ $x_{123}$ ） |
| | 细节展示（ $x_{13}$ ） | 有（ $x_{131}$ ） |
| | | 无（ $x_{132}$ ） |
| | 疑问和解答（ $x_{14}$ ） | 有（ $x_{141}$ ） |
| | | 无（ $x_{142}$ ） |
| | 搭配购买（ $x_{15}$ ） | 有（ $x_{151}$ ） |
| | | 无（ $x_{152}$ ） |

续表

| 区域位置 | 网页界面设计变量 | 变量水平 |
|---|---|---|
| 商品信息栏或相关信息栏 | 相关商品介绍位置（$x_{16}$） | 商品信息栏右侧（$x_{161}$） |
| | | 相关信息栏上部（$x_{162}$） |
| | | 相关信息栏下部（$x_{163}$） |

根据正交试验设计原理，运用 Adobe Dreamweaver 软件制作 32 个网页原型。基于网页原型和用户偏好进行用户情感体验主观测量实验，采用 7 级利克特量表（用户偏好得分区间为−3~3）进行评分，得到 32 个网页的平均用户偏好得分，如表 3 所示。

**表 3 用户偏好得分**

| WP | UP | WP | UP | WP | UP | WP | UP |
|---|---|---|---|---|---|---|---|
| $WP_{01}$ | −1.5000 | $WP_{09}$ | 0.5714 | $WP_{17}$ | −0.8571 | $WP_{25}$ | −0.5000 |
| $WP_{02}$ | 0.2857 | $WP_{10}$ | −0.5000 | $WP_{18}$ | 0.8750 | $WP_{26}$ | 0.0000 |
| $WP_{03}$ | 1.2857 | $WP_{11}$ | 0.8571 | $WP_{19}$ | −0.3750 | $WP_{27}$ | 0.2857 |
| $WP_{04}$ | −0.7143 | $WP_{12}$ | −0.5000 | $WP_{20}$ | −1.5000 | $WP_{28}$ | −0.7143 |
| $WP_{05}$ | 0.1250 | $WP_{13}$ | 0.8571 | $WP_{21}$ | 1.1250 | $WP_{29}$ | 1.0000 |
| $WP_{06}$ | 0.0000 | $WP_{14}$ | 1.0000 | $WP_{22}$ | −0.8571 | $WP_{30}$ | −1.7143 |
| $WP_{07}$ | −0.2500 | $WP_{15}$ | 0.1429 | $WP_{23}$ | 1.0000 | $WP_{31}$ | 1.0000 |
| $WP_{08}$ | 1.1429 | $WP_{16}$ | −0.1429 | $WP_{24}$ | −1.2857 | $WP_{32}$ | −1.3750 |

## 5.2 问题求解与讨论

1）问题求解

首先，将正交试验设计得到的 32 组不同的设计变量取值作为自变量集，把表 3 用户偏好得分作为因变量集，对数据进行标准化处理，带入式（3）~式（12）中进行计算，可以得到用户偏好与设计变量之间的函数关系。其次，运用表 1 所示的 IBPSO 步骤求解目标函数，设置种群规模 $G_{\max}=20$，最大迭代次数 $T_{\max}=300$，初始惯性权重 $\omega_s=0.9$，终止惯性权重 $\omega_e=0.4$，学习因子 $c_1=c_2=2$，计算结果如下：$x_{mn}^{*}=[10101001010101001010101010010100 1100]^{\mathrm{T}}$，即网站 logo 位置位于左侧、导航背景颜色与网页背景颜色不同色系（本文以橘红色为例）、小图辅助展示位置位于大图下部、商品名称位置位于大图上部、商品名称文字大小相对于其他信息文字应大且很明显、价格呈现形式应背景突出且字体有相应变化、有商品原价提醒、选择颜色时的颜色示例为商品附颜色示例且鼠标悬浮显示颜色名称、相关信息标签背景颜色与网页背景颜色不同色系（本文以橘红色为例）、有相关信息上部的店铺同类商品信息、有试穿体验或模特档案、模特展示和平铺展示屏数应该在 4 屏以下、有细节展示、有疑问和解答、没有搭配购买、相关商品介绍位置在商品信息栏右侧。

2）讨论

网页界面的导航背景颜色与网页背景颜色不同色系时，会提高用户的关注度，便于用户寻找和搜索信息。小图辅助展示位置位于大图下部时，可以方便用户进行切换浏览，而且绝大多数用户较为习惯此类设计。商品名称文字大小相对于其他信息文字大且很明显、价格呈现形式应背景突出且字体有相应变化、有商品原价提醒时，会使用户更加快捷地接收这些信息，减少信息收集时间。相关信息上部的店铺同类商品信息可以为用户提供同类商品的推荐，为用户提供更多的选择。有试穿体验或模特档案，可以

让用户较为方便地根据个人情况进行商品的尺码选择。模特展示和平铺展示屏数过多时，容易使用户产生烦躁的情绪，且有商品的细节展示，可以使用户在细节展示中查找所需要的商品信息，因此，模特展示和平铺展示只需要将商品全方位展示即可，不宜过多。有疑问和解答，可以让用户了解商品更多详细的信息，增加用户的信任度。

## 5.3 PLSR-IBPSO 算法有效性分析

1）PLSR 有效性分析

采用交叉有效性检验确定 PLSR 方程中选取成分的个数，计算结果如表 4 所示。

**表 4 交叉有效性检验**

| $h$ | $\theta^2$ | $\mathrm{Rd}(X;t_1,t_2,\cdots,t_h)$ | $\mathrm{Rd}(f(X);t_1,t_2,\cdots,t_h)$ | $Q_h^2$ | limit |
|---|---|---|---|---|---|
| 1 | 1.65 | 0.9583 | 0.9886 | 0.9823 | 0.0975 |
| 2 | 1.54 | 0.9896 | 0.9989 | 0.6180 | 0.0975 |
| 3 | 0.86 | 0.9899 | 0.9998 | −0.0932 | 0.0975 |

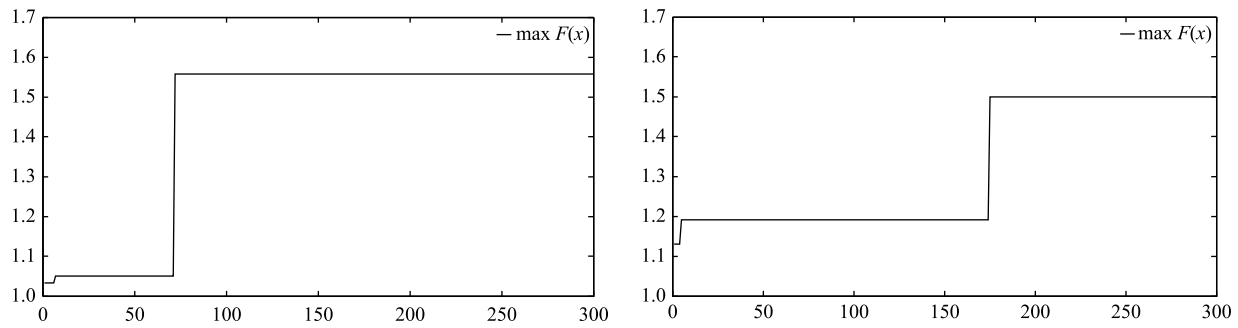
其中，$\theta^2$ 表示最大特征值；$\mathrm{Rd}(X;t_1,t_2,\cdots,t_h)$ 表示 $t_1,t_2,\cdots,t_h$ 对 $X$ 的累积解释能力；$\mathrm{Rd}(f(X);t_1,t_2,\cdots,t_h)$ 表示 $t_1,t_2,\cdots,t_h$ 对 $f(X)$ 的累积解释能力；$Q_h^2$ 表示交叉有效性检验值，其临界值为 0.0975，$Q_2^2=0.6180>0.0975$，$Q_3^2=-0.0932<0.0975$，因此选取两个成分即可，并且能够解释 99.89%的因变量信息，对自变量信息的利用率也达到 98.96%。

2）IBPSO 算法性能分析

为了验证 IBPSO 在求解有约束组合优化问题时的有效性，本文将其与 BPSO 进行比较，计算结果如表 5 和图 3 所示。其中，种群规模 $G_{\max}=20$，最大迭代次数 $T_{\max}=300$，初始惯性权重 $\omega_s=0.9$，终止惯性权重 $\omega_e=0.4$，学习因子 $c_1=c_2=2$。运行 100 次算法，统计平均运行时间（*N*-*t*/s）、平均收敛代数（*N*-iter）和用户偏好得分（*M*-CP）。实验用的 PC 配置为：3.4 GHz Inter（R）Core（TM）i7-6700 CPU，RAM 8.00 GB 安装内存。

**表 5 IBPSO 与 BPSO 的比较结果**

| 模型求解方法 | *N*-*t*/s | *N*-iter | *M*-CP |
|---|---|---|---|
| IBPSO | 10.8896 | 68 | 1.5578 |
| BPSO | 19.2368 | 172 | 1.4992 |




（a）IBPSO 算法最优个体适应度值演化过程　　（b）BPSO 算法最优个体适应度值演化过程

图 3 最优个体适应度值演化过程

由表 5 和图 3 可以得出，本文中提出的 IBPSO 在平均运行时间、平均收敛代数和用户偏好得分方

面都要优于传统的 BPSO。

3）仿真实验

为了进一步验证算法的改进效果，本文选取文献[30-34]中的计算实例作为本文的仿真实例。文献[30]的计算实例是有关网页界面设计优化问题，文献[31]是有关咖啡机外观设计优化问题，文献[32]是有门锁造型设计优化问题，文献[33]是有关功能手机外观设计优化问题，文献[34]是有关数码相机外观设计优化问题。分别用 IBPSO 和 BPSO 对仿真实例计算优化结果，并与文献中采用 GA 计算得到的优化方案进行比较，计算结果如表 6 所示。其中，种群规模 $G_{\max}=20$，最大迭代次数 $T_{\max}=300$，初始惯性权重 $\omega_s=0.9$，终止惯性权重 $\omega_e=0.4$，学习因子 $c_1=c_2=2$。运行 100 次算法，统计平均运行时间（*N*-*t*/s）、平均收敛代数（*N*-iter）和用户偏好得分（*M*-CP）。实验用的 PC 配置为：3.4 GHz Inter（R）Core（TM）i7-6700 CPU，RAM 8.00 GB 安装内存。

**表 6 IBPSO、BPSO 与 GA 的比较结果**

| 仿真实例 | IBPSO | | | BPSO | | | GA | | |
|---|---|---|---|---|---|---|---|---|---|
| | *N*-*t*/s | *N*-iter | *M*-CP | *N*-*t*/s | *N*-iter | *M*-CP | *N*-*t*/s | *N*-iter | *M*-CP |
| 文献[30] | 11.2946 | 43 | 1.9637 | 24.1967 | 50 | 1.6847 | 26.3742 | 51 | 1.5961 |
| 文献[31] | 10.2548 | 52 | 1.9654 | 15.2354 | 65 | 1.7548 | 14.9758 | 67 | 1.7257 |
| 文献[32] | 9.9825 | 47 | 1.8675 | 18.3645 | 136 | 1.6249 | 18.6952 | 129 | 1.5978 |
| 文献[33] | 13.1301 | 34 | 1.9986 | 18.6328 | 51 | 1.7437 | 19.1049 | 52 | 1.6986 |
| 文献[34] | 12.9525 | 79 | 1.8212 | 16.0214 | 208 | 1.6254 | 16.3952 | 212 | 1.6195 |

由表 6 可以得出，相较于传统的 BPSO 和 GA，本文中提出的 IBPSO 有较少的平均运行时间、较小的平均收敛代数和较高的用户偏好得分，并且传统的 BPSO 和 GA 在解决此类问题时，在算法收敛速度和最优解得分上差距不大。

## 6 结论

本文针对工程与管理领域中的有约束组合优化问题进行研究，即如何有效地对不同变量进行优化组合，以达到目标函数最优的要求。考虑变量维数大且变量间存在多重相关性的因素，以及变量间有强约束关系的特点，建立新的能有效反映该类问题实质的数学模型。提出运用 PLSR 构建有约束组合优化问题的目标函数，利用 IBPSO 对该类问题进行求解。制定了新的初始种群产生策略，保证在可行解空间内进行寻优；引入了动态惯性权重，保障算法具有更快的收敛性能；修改了种群更新机制并加入判别函数，确保种群在每次更新后都满足模型中的等式约束。算例和数值仿真分析结果表明，本文提出的基于 PLSR-IBPSO 的方法，相较于传统方法有较高的预测精度和较快的收敛速度，计算结果更稳定。研究结果表明，该方法可以有效解决工程与管理领域中的有约束组合优化问题。

## 参 考 文 献

[1] Blum C，Pinacho P，López-Ibáñez M，et al. Construct，merge，solve & adapt a new general algorithm for combinatorial optimization[J]. Computers & Operations Research，2016，68：75-88.

[2] Martí R，Resende M G C，Ribeiro C C. Multi-start methods for combinatorial optimization[J]. European Journal of Operational Research，2013，226（1）：1-8.

[3] 王军，李端. 多项式 0-1 规划中隐枚举算法的改进及应用[J]. 系统工程理论与实践，2007，27（3）：21-27.
[4] Heilmann R. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags[J]. European Journal of Operational Research，2003，144（2）：348-365.
[5] 赵晓煜，汪定伟. 供应链中二级分销网络的优化设计模型[J]. 管理科学学报，2001，4（4）：22-26，72.
[6] Gomory R E. Outline of an algorithm for integer solutions to linear programs[J]. Bulletin of the American Mathematical Society，1958，64（5）：275-278.
[7] Gilmore P C，Gomory R E. A linear programming approach to the cutting-stock problem[J]. Operations Research，1961，9（6）：849-859.
[8] Gilmore P C，Gomory R E. A linear programming approach to the cutting stock problem-part II[J]. Operations Research，1963，11（6）：863-1025.
[9] Blazewicz J，Lenstra J K，Rinnooy-Kan A H G. Scheduling subject to resource constraints：classification and complexity[J]. Discrete Applied Mathematics，1983，5（1）：11-24.
[10] Zamani R. A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem[J]. European Journal of Operational Research，2013，229（2）：552-559.
[11] Gonçalves J F，Resende M G C，Mendes J J. A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem[J]. Journal of Heuristics，2011，17（5）：467-486.
[12] 游晓明，刘升，吕金秋. 一种动态搜索策略的蚁群算法及其在机器人路径规划中的应用[J]. 控制与决策，2017，32(3)：552-556.
[13] Bououden S，Chadli M，Karimi H R. An ant colony optimization-based fuzzy predictive control approach for nonlinear processes[J]. Information Sciences，2015，299：143-158.
[14] Bouleimen K，Lecocq H. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version[J]. European Journal of Operational Research，2003，149（2）：268-281.
[15] Wang C，Mu D，Zhao F，et al. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows[J]. Computers & Industrial Engineering，2015，83：111-122.
[16] Nonobe K，Ibaraki T. Formulation and tabu search algorithm for the resource constrained project scheduling problem[C]//Ribeiro C C，Hansen P. Essays and Surveys in Metaheuristics. Boston：Springer，2002：557-588.
[17] 胡祥培，孙丽君，王雅楠. 物流配送系统干扰管理模型研究[J]. 管理科学学报，2011，14（1）：50-60.
[18] Xu X L，Rong H Z，Trovati M，et al. CS-PSO：chaotic particle swarm optimization algorithm for solving combinatorial optimization problems[J]. Soft Computing，2018，22（3）：783-795.
[19] Elloumi W，El Abed H，Abraham A，et al. A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP[J]. Applied Soft Computing，2014，25：234-241.
[20] Kennedy J，Eberhart R C. Particle swarm optimization[C]. New York：1995 IEEE International Conference on Neural Networks.
[21] Shi Y，Eberhart R. A modified particle swarm optimizer[C]. Anchorage：1998 IEEE International Conference on Evolutionary Computation.
[22] Angeline P J. Using selection to improve particle swarm optimization[C]. Anchorage：1998 IEEE International Conference on Evolutionary Computation.
[23] Angeline P J. Evolutionary optimization versus particle swarm optimization：philosophy and performance differences[C]. San Diego：International Conference on Evolutionary Programming，1998.
[24] Kennedy J，Mendes R. Population structure and particle swarm performance[C]. Honolulu：2002 Congress on Evolutionary Computation.
[25] Suganthan P N. Particle swarm optimiser with neighbourhood operator[C]. Washington：1999 Congress on Evolutionary Computation.
[26] Parsopoulos K E，Vrahatis M N. Recent approaches to global optimization problems through particle swarm optimization[J]. Natural computing，2002，1（2/3）：235-306.
[27] 谭瑛，高慧敏，曾建潮. 求解整数规划问题的微粒群算法[J]. 系统工程理论与实践，2004，24（5）：126-129.
[28] Kennedy J，Eberhart R C. A discrete binary version of the particle swarm algorithm[C]. Orlando：1997 IEEE International Conference on Systems，Man，and Cybernetics.
[29] 吴登生，李建平，蔡晨. 软件成本估算的粒子群算法类比模型及自助法推断[J]. 管理科学，2010，23（3）：113-120.
[30] 郭伏，李美林，屈庆星. 基于感性工学的电子商务网页外观设计优化[J]. 人类工效学，2013，19（3）：56-60.
[31] Hsiao S W，Chiu F Y，Lu S H. Product-form design model based on genetic algorithms[J]. International Journal of Industrial Ergonomics，2010，40（3）：237-246.
[32] Hsiao S W，Tsai H C. Applying a hybrid approach based on fuzzy neural network and genetic algorithm to product form

design[J]. International Journal of Industrial Ergonomics，2005，35（5）：411-428.
[33] Guo F，Qu Q X，Chen P，et al. Application of evolutionary neural networks on optimization design of mobile phone based on user's emotional needs[J]. Human Factors and Ergonomics in Manufacturing & Service Industries，2016，26（3）：301-315.
[34] Guo F，Liu W L，Liu F T，et al. Emotional design method of product presented in multi-dimensional variables based on Kansei Engineering[J]. Journal of Engineering Design，2014，25（4/5/6）：194-212.

## The PLSR-IBPSO Algorithm for Solving Constrained Combinatorial Optimization Problem—A Case Study of User Interface Design

GUO Fu，QU Qingxing
（School of Business Administration，Northeastern University，Shenyang 110167，China）

**Abstract** Analyzing the constrained combinatorial optimization problem in engineering and management fields，considering large-scaled and correlated variables with the equality constraint，a novel mathematical model has been proposed to fit actual problem optimization. The partial least squares regression（PLSR）was used to build relationship model between independent and dependent variables，and the improved binary particle swarm optimization（IBPSO）algorithm has been developed to seek optimum value. And the IBPSO algorithm has been developed for solving the problem how to arrange n levels in m variables to seek optimum target function value. In the IBPSO，a new method of making initial particles has been presented for searching for optimum particle in the feasible dimensional problem space. Furthermore，a dynamic inertia weight was importing in algorithm to expedite convergence speed. Finally，a modified update mechanism has been used for making updated particles to meet the equality constraint of mathematical model. Algorithm examples research demonstrates that the IBPSO algorithm is effective and can achieve good results.

**Key words** Combinatorial optimization，Partial least squares regression，Binary particle swarm optimization，Constrained optimization，Product design optimization

### 作者简介

郭伏（1964—），女，东北大学工商管理学院教授、博士生导师，研究方向：用户体验、人因工程、感性工学与情感化设计等。E-mail：fguo@mail.neu.edu.cn。

屈庆星（1988—），男，东北大学工商管理学院博士研究生，研究方向：用户体验、人因工程、感性工学与情感化设计等。E-mail：yantaiquqingxing@163.com。