第 29 卷第 10 期　　计算机应用　　Vol. 29 No. 10
2009 年 10 月　　Journal of Computer Applications　　Oct. 2009

文章编号:1001 - 9081(2009)10 - 2762 - 04

一类带延迟策略的库存优化模型及其仿真

刘兵兵
(安庆师范学院 数学与计算科学学院,安徽 安庆 246133)
(lbb122400@gmail.com)

摘　要:考虑一类带延迟策略的库存优化模型，即二层整数规划问题。证明了该二层整数规划问题等价于约束单层整数规划问题。借助罚函数思想化约束整数规划问题为无约束整数规划问题，再利用遗传算法进行求解。数值模拟表明所得数值结果与已有的数值结果相比,不仅使得供应链整体库存效益有较大提高，并且对每个库存分点的最优库存量作了更为合理的调整。

关键词:多级库存;延迟策略;二层整数规划;遗传算法;最优解
中图分类号: TP18　　**文献标志码**:A

Inventory optimization model with postponement strategy and its simulation

LIU Bing-bing
(*School of Mathematics and Computing Science, Anqing Teachers College, Anqing Anhui* 246133, *China*)

Abstract: In this paper, a sort of optimal inventory model with postponement strategy i. e. bilevel integer programming problem was researched. Bilevel integer programming problem was proved to be equivalent to constrained integer programming problem, and could be transformed to integer programming problem without constraint via penalty function. A genetic algorithm was proposed to solve this problem. The numerical simulation results show that the proposed model can improve the inventory benefits of overall supply chain, and reasonably adjust the optimal inventory of all branch points.

Key words: multi-echelon inventory; postponement strategy; bilevel integer programming; Genetic Algorithm (GA); optimal solution

0　引言

多级库存优化问题是物流供应链管理理论和实践当中一个重要议题,它制约着供应链整体性能的提高。许多学者对该问题进行了多方面的探讨:文献[1 - 2]作者对此作了较全面的综述;文献[3 - 4]作者分别就多级库存系统的紧急补救策略和一种随机需求的多级库存策略作了较详细的研究;文献[5]作者引入多目标规划理论,建立了同时考虑需求满足率、时间和成本等多个目标的协同库存优化模型,并借助二层规划理论和进化多目标优化技术设计了协同多级库存的数学模型以及模型的求解算法;文献[6]作者提出了一种集中式和分布式库存共存的带延迟策略的混合库存模型,同时也借助二层规划理论和混沌优化技术设计了相关模型和求解算法。但该文也提出了一个公开问题,即可以设计更好的求解算法进一步改善模型的数值结果,进而更高效地去服务生产企业。

1　带延迟策略的库存优化模型

所谓延迟,即只有当企业接受了客户的订单,才进行最终的生产制造。在该过程中,带有客户个性特征的原材料的组合能够保证产品达到大批量订货,它可以快速、低成本且高效地满足客户的需要。因此,该策略能够将规模化和个性化两个表面上看似矛盾的概念有机地组合起来[7]。

供应链中带延迟策略的库存批量订货在实际中经常遇到,它决定着生产企业如何去选择合适的库存模型:集中式的库存,分布式的库存还是两种形式并存的混合库存。文献[8]作者认为集中式的库存可以为采取延迟策略的企业带来更多的利润,但实现完全的延迟策略存在诸多的实际困难,而完全的延迟策略实际上并不是一个最佳决策。因此文献[6]作者提出了一个带延迟策略的混合库存模型,并提出用二层规划模型去优化供应链整体库存效益和各分点的库存效益,其模型如下:

$$(BP)_U:\ \mathrm{Max}\ F = \sum_{i=0}^{n} F_i(x^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)})$$

$$\text{s.t.}\begin{cases}\alpha^{(i)} \leqslant x^{(i)} \leqslant \beta^{(i)}\\ \alpha \leqslant \sum\limits_{i=0}^{n} x^{(i)} \leqslant \beta\\ x^{(i)} \geqslant 0;\ x^{(i)} \in \mathbf{Z},\ i = 0,1,2,\cdots,n\end{cases}\tag{1}$$

$$(BP)_L^i:\ \mathrm{Max}\ F_i(x^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)}) = f_i(x^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)}$$

$$\text{s.t.}\begin{cases}c_1^{(i)} \leqslant y_1^{(i)} \leqslant d_1^{(i)}\\ c_2^{(i)} \leqslant y_2^{(i)} \leqslant d_2^{(i)}\\ c_3^{(i)} \leqslant y_3^{(i)} \leqslant d_3^{(i)}\\ y_1^{(i)} + y_2^{(i)} + y_3^{(i)} \leqslant x^{(i)}\\ y_j^{(i)} \geqslant 0;\ y_j^{(i)} \in \mathbf{Z}(i = 0,1,2,\cdots,n;\ j = 1,2,3)\end{cases}\tag{2}$$

收稿日期:2009 - 04 - 22;**修回日期**:2009 - 06 - 22。　　**基金项目**:安徽省高校优秀青年人才基金项目(2009SQRZ121)。
作者简介:刘兵兵(1980 -),男,河北武安人,讲师,硕士,主要研究方向:数学规划、遗传算法。

其中：$(BP)_U$ 为供应链整体库存效益模型，$(BP)_L^i$ 为各分库库存效益模型。α 为库存的最小存量，β 为库存的上限，二者都是通过对市场需求预测得到的。$x^{(i)}$ 为上层决策变量，表示各分库的库存量；$x^{(0)}$ 为总部的库存量。F 为供应链整体的库存效益函数，而 $F_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)})$ 表示各库存分点的效益函数（当 $i=0$ 时，它表示总部的效益，即为集中式的库存效益），$\sum_{i=0}^{n} x^{(i)}$ 表示供应链整体的库存量。$y_1^{(i)},y_2^{(i)},y_3^{(i)}$ 分别表示各库存分点的原料库存量、半成品库存量和成品库存量。$f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)})$ 表示各库存分点的效益函数，$\sum_{j=1}^{3} e_j y_j^i$ 表示各库存分点的成本函数。e_1,e_2,e_3 分别表示原料、半成品和成品的平均成本。$c_1^{(i)},d_1^{(i)}$ 表示各库存分点的原料存量的上下限；$c_2^{(i)},d_2^{(i)}$ 表示各库存分点的半成品存量的上下限；$c_3^{(i)}$，$d_3^{(i)}$ 表示各库存分点的成品存量的上下限。

2　模型的分析与简化

一般地，在已有文献中关于多级库存管理模型数学本质上多为单随从多层规划问题，而其中又以单随从二层规划问题最多。通过求解此二层规划问题以寻求成品库存的最优控制。而在带延迟策略的库存模型中，通过对原材料、半成品和成品的库存量进行最优控制，既可满足客户对产品的特殊要求，又可以使得生产商达到批量生产，在此基础上使得库存效益最大化。本文所研究的带延迟策略的多级库存模型，本质上为多随从二层规划问题，即上层为独立目标函数，下层为若干个平行的目标函数，且下层各目标函数的决策变量互不影响，彼此独立。因此在下面的分析中，借助了独立多随从二层规划问题的相关理论。

把上述二层规划模型（1）～（2）记为 BP（Bilevel Programming）问题，可以证明该 BP 问题其实等价于单层整数规划（Integer Programming，IP），即为：

$$\text{Max } \bar{F} = \sum_{i=0}^{n} \left(f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)} \right)$$

$$\text{s. t.} \begin{cases} \alpha^{(i)} \leqslant x^{(i)} \leqslant \beta^{(i)} \\ \alpha \leqslant \sum_{i=0}^{n} x^{(i)} \leqslant \beta \\ c_1^{(i)} \leqslant y_1^{(i)} \leqslant d_1^{(i)} \\ c_2^{(i)} \leqslant y_2^{(i)} \leqslant d_2^{(i)} \\ c_3^{(i)} \leqslant y_3^{(i)} \leqslant d_3^{(i)} \\ y_1^{(i)} + y_2^{(i)} + y_3^{(i)} \leqslant x^{(i)} \\ x^{(i)} \geqslant 0 \ (x^{(i)} \in \mathbf{Z},\ i = 0,1,2,\cdots,n) \\ y_j^{(i)} \geqslant 0 \ (y_j^{(i)} \in \mathbf{Z};i = 0,1,2,\cdots,n,j = 1,2,3) \end{cases} \tag{3}$$

在给出证明过程之前，先给出关于 BP 问题和 IP 问题的几个相关定义。

定义 1　BP 问题的约束集为：

$$S_{\mathrm{BP}} = \left\{ (x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) \mid \begin{cases} \alpha^{(i)} \leqslant x^{(i)} \leqslant \beta^{(i)} \\ \alpha \leqslant \sum_{i=0}^{n} x^{(i)} \leqslant \beta \\ x^{(i)} \geqslant 0(x^{(i)} \in \mathbf{Z}) \end{cases}, \begin{cases} c_1^{(i)} \leqslant y_1^{(i)} \leqslant d_1^{(i)} \\ c_2^{(i)} \leqslant y_2^{(i)} \leqslant d_2^{(i)} \\ c_3^{(i)} \leqslant y_3^{(i)} \leqslant d_3^{(i)} \\ y_1^{(i)} + y_2^{(i)} + y_3^{(i)} \leqslant x^{(i)} \\ y_j^{(i)} \geqslant 0(y_j^{(i)} \in \mathbf{Z}) \end{cases}, i = 0,1,\cdots,n \right\} \tag{4}$$

并设 S_{BP} 为非空紧集。

定义 2　对于给定的上层决策变量 x，第 i 个随从的可行集为：

$$S_{\mathrm{BP}}^i(x^{(i)}) = \left\{ (y_1^{(i)},y_2^{(i)},y_3^{(i)}) \mid \begin{cases} c_1^{(i)} \leqslant y_1^{(i)} \leqslant d_1^{(i)} \\ c_2^{(i)} \leqslant y_2^{(i)} \leqslant d_2^{(i)} \\ c_3^{(i)} \leqslant y_3^{(i)} \leqslant d_3^{(i)} \\ y_1^{(i)} + y_2^{(i)} + y_3^{(i)} \leqslant x^{(i)} \\ y_j^{(i)} \geqslant 0(y_j^{(i)} \in \mathbf{Z}) \end{cases}, i = 0,1,\cdots,n \right\} \tag{5}$$

定义 3　S_{BP} 在上层决策空间的投影：

$$S_{\mathrm{BP}}(x) = \left\{ (x^{(i)}) \mid \exists (y_1^{(i)},y_2^{(i)},y_3^{(i)}), \begin{cases} \alpha^{(i)} \leqslant x^{(i)} \leqslant \beta^{(i)} \\ \alpha \leqslant \sum_{i=0}^{n} x^{(i)} \leqslant \beta \\ x^{(i)} \geqslant 0(x^{(i)} \in \mathbf{Z}) \end{cases}, \begin{cases} c_1^{(i)} \leqslant y_1^{(i)} \leqslant d_1^{(i)} \\ c_2^{(i)} \leqslant y_2^{(i)} \leqslant d_2^{(i)} \\ c_3^{(i)} \leqslant y_3^{(i)} \leqslant d_3^{(i)} \\ y_1^{(i)} + y_2^{(i)} + y_3^{(i)} \leqslant x^{(i)} \\ y_j^{(i)} \geqslant 0(y_j^{(i)} \in \mathbf{Z}) \end{cases}, i = 0,1,\cdots,n \right\} \tag{6}$$

定义 4　对于 $x^{(i)} \in S_{\mathrm{BP}}(x)$，第 i 个随从的合理反应集为：

$$P^{(i)}(x^{(i)}) = \underset{y_1^{(i)},y_2^{(i)},y_3^{i}}{\operatorname{argmin}} \left\{ -f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) + \sum_{j=1}^{3} e_j y_j^{(i)} \mid (y_1^{(i)},y_2^{(i)},y_3^{(i)}) \in S_{\mathrm{BP}}^i(x^{(i)}) \right\} \tag{7}$$

在这里假设 $P^{(i)}(x^{(i)})(i=0,1,\cdots,n)$ 为非空的，即对上层作出的所有决策，每个下层都有自己的决策空间，并且保证 $P^{(i)}(x^{(i)})(i=0,1,\cdots,n)$ 为点对点影射[9-10]。

定义 5　上层问题的可行集为：

$$IR_{\mathrm{BP}} = \{ (x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) \mid (x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) \in S_{\mathrm{BP}}, (y_1^{(i)},y_2^{(i)},y_3^{(i)}) \in P^{(i)}(x^{(i)}), i = 0,1,\cdots,n \} \tag{8}$$

为了保证 BP 问题的解存在，在这里假设 IR_{BP} 非空。

定义 6　IP 问题的可行集为：

$$IR_{\mathrm{IP}} = \{ (x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) \mid (x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) \text{ 满足约束(6)} \} \tag{9}$$

定理 1　BP 问题与 IP 问题是等价的。

证明　设 BP 问题已得到最优解为 $(\bar{x}^{(i)},\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)})(i=0,1,\cdots,n)$，则 $(\bar{x}^{(i)},\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)}) \in IR_{\mathrm{BP}}$，因此，$(\bar{x}^{(i)},\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)}) \in S_{\mathrm{BP}}$ 且 $(\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)}) \in P^{(i)}(\bar{x}^{(i)})(i=0,1,\cdots,n)$。由定义 4 可知 $(\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)})(i=0,1,\cdots,n)$ 使得 n 个随从的目标函数都达到最优，即 $f_i(x^{(i)},y_1^{(i)},y_2^{(i)},$

$y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)} (i = 0,1,\cdots,n)$ 都取得最优值。那么也就意味着 $\sum_{i=0}^{n} \left(f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)}\right)$ 取得最优值,即 $(\bar{x}^{(i)},\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)})(i = 0,1,\cdots,n)$ 使得IP问题取得最优值,并且 $(\bar{x}^{(i)},\bar{y}_1^{(i)},\bar{y}_2^{(i)},\bar{y}_3^{(i)}) \in IR_{\mathrm{IP}}(i = 0,1,\cdots,n)$。

反之,设IP问题已得到最优解为 $(\tilde{x}^{(i)},\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)})(i = 0,1,\cdots,n)$,也即意味着它使得每个 $f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)} (i = 0,1,\cdots,n)$ 都达到最优,因此也使得BP问题的上层决策目标函数(1)取得最优值,且有 $(\tilde{x}^{(i)},\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)}) \in IR_{\mathrm{IP}}(i = 0,1,\cdots,n)$。因此,$(\tilde{x}^{(i)},\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)}) \in S_{\mathrm{BP}}(i = 0,1,\cdots,n)$。对于给定的 $\tilde{x}^{(i)}$,$(\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)})$ 使得BP问题的每个下层目标函数 $f_i(x^{(i)},y_1^{(i)},y_2^{(i)},y_3^{(i)}) - \sum_{j=1}^{3} e_j y_j^{(i)} (i = 0,1,\cdots,n)$ 取得最优,且有 $(\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)}) \in P^{(i)}(\tilde{x}^{(i)})(i = 0,1,\cdots,n)$ 成立。由此可得 $(\tilde{x}^{(i)},\tilde{y}_1^{(i)},\tilde{y}_2^{(i)},\tilde{y}_3^{(i)}) \in IR_{\mathrm{BP}}$。综上两方面,定理得证。

3 求解算法的设计

由于BP问题的非凸、非连续性和内在嵌套性,求解该问题是一个NP难问题[11-12]。因此,本文专门设计了一类遗传算法求解IP问题,来代替求解BP问题。如果上述第2部分IP问题的目标函数为非线性函数,则要求设计求解非线性整数规划问题的算法,比如传统的分支定界算法和线性逼近法等[13],但是这些传统的算法对于较大规模的非线性IP问题求解起来耗时很多,不具有实时性。遗传算法[14]是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型,是一种新的全局优化搜索算法,它简单通用,鲁棒性强,适于并行处理,已经广泛地应用在计算机科学、优化调度、运输问题、组合优化等领域。下面给出遗传算法的具体求解步骤。

3.1 染色体编码方式

由于变量为整数,理想的编码方式为二进制编码,但是二进制编码会产生“海明悬崖”,导致最终解的不稳定性,因此本文使用实数编码。

3.2 适应值函数

使用函数思想,把有约束的IP问题转化成求极小的无约束的IP问题,再把无约束的IP问题的目标函数作为适应值函数。

3.3 交叉策略

采用多点交叉算子,把个体编码为浮点表示的向量并不排除一点或多点交叉,因为这些杂交算子并不依赖于二进制表示,只要求作用的对象是某种列的表示,所以它们同样可以用在浮点表示的个体上,但杂交位置只允许选各个分量之间处,因此把决策变量表示成向量的形式。以一定概率选取两个父代个体 $X_i = (x_{i1},x_{i2},\cdots,x_{in})$ 和 $X_j = (x_{j1},x_{j2},\cdots,x_{jn})$,$n$ 为决策变量的个数,在1到 n 之间随机地选择两个整数 k_1,$k_2(k_1 < k_2)$,例如 X_1,X_2 为两个长 $n = 9$ 的整数串组成的染色体,选取 $k_1 = 3$,$k_2 = 7$,交换得到新的染色体 X'_1,X'_2。

$$\begin{cases} X_1 = x_{11}x_{12}x_{13} \mid x_{14}x_{15}x_{16} \mid x_{17}x_{18}x_{19} \\ X_2 = x_{21}x_{22}x_{23} \mid x_{24}x_{25}x_{26} \mid x_{27}x_{28}x_{29} \end{cases}$$

变换后为:

$$\begin{cases} X'_1 = x_{11}x_{12}x_{13} \mid x_{24}x_{25}x_{26} \mid x_{17}x_{18}x_{19} \\ X'_2 = x_{21}x_{22}x_{23} \mid x_{14}x_{15}x_{16} \mid x_{27}x_{28}x_{29} \end{cases}$$

3.4 变异策略

采用均匀变异算子,若个体为 $X_i = (x_{i1},x_{i2},\cdots,x_{in})$,则每个分量 x_{ik} 以完全相同的概率进行变异,一次变异后的结果为 $(x_{i1},x_{i2},\cdots,x'_{ik},\cdots,x_{in})(1 \leqslant k \leqslant n)$,其中 x'_{ik} 使第 k 个参变量定义域中的一个随机值。

3.5 选择策略

采用锦标赛选择算子[15]并将整个种群分为 l 组子种群,将上一代的 m 个最优解替换每个子种群 m 个最差的个体,其余个体不变。

3.6 最优保留策略

最优的个体不一定出现在最后一代,所以在进化开始时就必须把最好的个体保留下来,若在新种群中发现更好的个体,则用它代替前面保留的个体,在进化完成后,保留下来的个体就看作问题的最优解。

3.7 终止条件

设 G 为最大代数且连续进化 n 代种群没有改进,算法停止,输出最优解。

3.8 算法时间复杂度及收敛性分析

所给算法的时间复杂度分析应分成两部分来计算:第一部分是种群初始化,由于种群初始化是在所给问题的基本空间中随机生成的,故其时间复杂度为 $O(N)$,其中 N 为种群规模;第二部分是三种遗传操作。在选择操作中,采用的是锦标赛选择算子,其选择机理为随机地在群体中选择 k 个个体进行比较,适应度最好的个体将被选择复制到下一代,参数 k 称为竞赛规模,本文中取 $k = 2$,故其时间复杂度为 $O(GN)$,其中 N 为种群规模,G 为遗传进化终止代数。在交叉操作中,采用的是两点交叉算子,故其时间复杂度为 $O(GN)$。在变异操作中,采用的是均匀变异算子,时间复杂度为 $O(GN)$,所以整个算法的时间复杂度为 $O(GN)$,它是一个线性时间算法。

本文所给算法采取了最优保留策略,文献[16]中证明了采用最优保留策略的遗传算法能收敛于最优解的概率为1。另外,我们采用了多子群并行搜索的策略。由此可见,该算法在计算时间和收敛速度上具有较高的性能,能够在短时间内找到问题的最优解或较优解。

4 仿真测试

为了验证算法的有效性,用Matlab语言编制了仿真程序

并在配置为 AMD Athlon(tm)64 Processor、频率为 2.01 GHz 的 CPU、512 MB 内存的 PC 机上进行了测试。算例来自于文献[6],该问题由定理 1 可以等价为下面的 IP 问题,即为:

$$\begin{aligned}\mathrm{Max}_x \bar{F}(x,y) = {} & 60 + 1/(0.0016 - 17.9858\mathrm{e}^{-x^{(0)}} + \\ & 0.0008\mathrm{e}^{-y_1^{(0)}} + 0.3886\mathrm{e}^{-y_2^{(0)}} + 0.0003e^{-y_3^{(0)}}) - \\ & 1.782y_1^{(0)} - 1.034y_2^{(0)} - 0.921y_3^{(0)} + \\ & 1/(0.0135 - 0.8599\mathrm{e}^{-x^{(1)}} + 0.0173\mathrm{e}^{-y_1^{(1)}} + \\ & 0.1182\mathrm{e}^{-y_2^{(1)}} + 0.0099\mathrm{e}^{-y_3^{(1)}}) - \\ & 1.987y_1^{(1)} - 1.157y_2^{(1)} - 1.015y_3^{(1)} + \\ & 1/(0.0097 - 4.2173\mathrm{e}^{-x^{(2)}} + 0.0156\mathrm{e}^{-y_1^{(2)}} + \\ & 0.0357\mathrm{e}^{-y_2^{(2)}} + 1.0364\mathrm{e}^{-y_3^{(2)}}) - \\ & 2.013y_1^{(2)} - 1.228y_2^{(2)} - 1.201y_3^{(2)}\end{aligned}$$

$$\begin{aligned}\mathrm{s.t.}\ & 11 \leqslant x^{(0)} \leqslant 22, 4 \leqslant x^{(1)} \leqslant 15, 6 \leqslant x^{(2)} \leqslant 17, 21 \leqslant \sum_{i=0}^{2} x^{(i)} \leqslant 54, 1 \leqslant y_1^{(0)} \leqslant 8, 8 \leqslant y_2^{(0)} \leqslant 11, 2 \leqslant y_3^{(0)} \leqslant 3, y_1^{(0)} \\ & + y_2^{(0)} + y_3^{(0)} \leqslant x^{(0)}, 1 \leqslant y_1^{(1)} \leqslant 5, 2 \leqslant y_2^{(1)} \leqslant 7, 1 \leqslant y_3^{(1)} \leqslant 3, y_1^{(1)} + y_2^{(1)} + y_3^{(1)} \leqslant x^{(1)}, 1 \leqslant y_1^{(2)} \leqslant 4, 1 \leqslant y_2^{(2)} \leqslant 5, 4 \leqslant \\ & y_3^{(2)} \leqslant 8, y_1^{(2)} + y_2^{(2)} + y_3^{(2)} \leqslant x^{(2)}, x^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)} \in \mathbf{Z}(i = 0,1,2)\end{aligned}$$

算法所用参数如下,种群规模 $N = 80$,遗传进化终止代数 $G = 100$,种群进化停滞的最大代数 $n = 30$,交叉概率 $p_c = 0.7$,变异概率 $p_m = 0.35$。

表 1　本文的仿真结果与文献[6]中的结果比较

算法	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$y^{(0)}$	$y^{(1)}$	$y^{(2)}$	F	F_0	F_1	F_2
文献[6]算法	11	5	10	1 8 2	1 3 2	1 2 7	685.24	605.89	50.447	48.075
本文算法	16	9	11	4 9 3	2 5 2	3 2 6	729.0436	612.6321	59.1622	57.2453

运行该程序 10 次,算法运行占用 CPU 平均时间约为 200 ms。

从所得结果比较来看,本文的实验结果所得的供应链整体库存效益为 729.0436,要比文献[6]中的供应链整体库存效益 685.24 提高很多,主要原因是各库存分点对原材料、半成品和成品的最优库存量进行了更为科学合理的调整。

5　结语

本文在文献[6]给出的两级库存二层整数规划模型的基础上,首先证明该二层整数规划等价于约束整数规划问题,然后通过罚函数思想转化为无约束整数规划问题,设计了一类遗传算法,数值实验结果显示该算法不仅对于供应链整体库存效益有所提高,而且对各个库存分点的三种形式库存量的最优值作了较为合理的调整,算法运行时间短,实时性较强。

参考文献:

[1] ZIPKIN P H. Foundation of Inventory Management[M]. New York: Mcgraw Hill Higher Education, 2000: 109 -110.

[2] SIMCHI-LEVI D, CHEN XIN, BRAMEL J. The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management [M]. New York: Springer-Verlag, 2004: 21 -58.

[3] CHUI H N, HUANG H L. A multi-ecelon integrated JIT inventory model using the time buffer and emergency borrowming policies to deal with randon delivery lead times[J]. International Journal of Production Research, 2003, 41(13): 2911 -2931.

[4] GANESHAN R. Managing supply chain inventories: A multiple supplier, one warehouse, multiple retailer model[J]. International Journal of Production Economics, 1999, 59(2):341 -354.

[5] 卫忠, 徐晓飞, 战德臣, 等. 协同供应链多级库存控制的多目标优化模型以及求解方法[J]. 自动化学报, 2007, 33(2): 181 -187.

[6] GUI S P, NIU B Z, ZHANG Z Y, *et al.* An optimal inventory model based on postponement strategy: A bi-level programming approach [C]// 2007 Second IEEE Conference on Industrial Electronics and Applications. [S. l.]: IEEE, 2007: 446 -450.

[7] DAVID M , ANDERSON P E . Implementing mass customization [J]. Agility and Global Competition, 1998, 2(2): 36 -49.

[8] CHOPRA S, MEINDL P. Supply chain Management: Strategy, Planning, and Operation[M]. Upper Saddle River: Prentice-Hall Inc, 2001.

[9] BARD J F. Practical Bilevel Optimization: Algorithms and Applications[M]. London: Kluwer, 1998: 10 -30.

[10] DEMPE S. Foundations of Bilevel Programming[M]. London: Kluwer, 2002: 15 -32.

[11] BARD J F. Some properties of the bilevel linear programming[J]. Journal of Optimization Theory and Applications, 1991, 68(2): 146 -164.

[12] BLAIR B-A O. Computational difficulties of bilevel linear programming[J]. Operations Research, 1990, 38(3): 556 -560.

[13] LI D, SUN X L. Nonlinear Integer Programming[M]. New York: Springer, 2006: 61 -89.

[14] HOLLAND J H. Adaptation in Natural Systems[M]. Ann Arbor: The University of Michigan Press, 1975: 5 -13.

[15] GOLDBERG D E. Real-coded genetic algorithms, virtual alphabets, and blocking, Technical Report, No.90001[R]. Champaign: University of Illinois at Urbana-Champaign, 1990.

[16] 徐宗本, 张讲社, 郑亚林. 计算智能中的仿生学: 理论与算法[M]. 北京:科学出版社, 2003: 69 -77.