

Modeling and optimization of unbalanced multi-stage logistic system

Xu Hang Xu Rong Ye Qingtai

(School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: To decompose an unbalanced multi-stage logistic system to multiple independent single-stage logistic systems, a new notion of parameterized interface distribution is presented. For encoding the logistic pattern on each stage, the Prüfer number is used. With the improved decoding procedure, any Prüfer number produced stochastically can be decoded to a feasible logistic pattern, which can match with the capacities of the nodes of the logistic system. With these two innovations, a new modeling method based on parameterized interface distribution and the Prüfer number coding is put forward. The corresponding genetic algorithm, named as PIP-GA, can find better solutions and require less computational time than st-GA. Although requiring a little more consumption of memory, PIP-GA is still an efficient and robust method in the modeling and optimization of unbalanced multi-stage logistic systems.

Key words: logistics; optimization; genetic algorithm; Prüfer number; spanning tree; parameterized interface distribution; unbalanced multi-stage logistic system

If several facilities are to be sited between the suppliers and the customers, such as plants or regional distribution centers (DCs), then the multi-stage model is the appropriate model^[1,2]. Multi-stage logistic system design is a typical difficult problem in the management of logistic systems.

The use of the spanning tree-based genetic algorithm (st-GA) for solving multi-stage logistic system problems was introduced by Gen et al^[3,4]. They employed the Prüfer number to represent a candidate solution of the problems and developed feasibility criteria of the Prüfer number to be decoded into a spanning tree. In Ref. [1], the st-GA is improved.

In this paper, the notion of parameterized interface distribution is put forward, by which an unbalanced logistic system can be converted to a set of balanced ones. Meanwhile, with the improved decoding procedure, any Prüfer number produced stochastically can be decoded to a feasible logistic pattern, which can match with the capacities of the nodes of the logistic system.

With these two innovations, a new genetic algorithm based on parameterized interface distribution and Prüfer number coding (PIP-GA) comes into being. The efficacy and the efficiency of PIP-GA are demonstrated by comparing its numerical experimental result with that of the st-GA presented in Ref. [1].

1 Unbalanced Multi-Stage Logistic System

Suppose that there are m suppliers and n customers in a logistic stage.

First, we denote $s = \{1, 2, \dots, m\}$ as the supplier vector, $d = \{m + 1, m + 2, \dots, m + n\}$ as the customer vector. Secondly, the capacity of the supplier or the customer is denoted as $a_i, i \in s \cup d$; we denote $S = \{a_1, a_2, \dots, a_m\}$ as the supplier capacity vector, and denote $D = \{a_{m+1}, a_{m+2}, \dots, a_{m+n}\}$ as the customer capacity vector. Thirdly, denote $|S| = \sum_{i=1}^m a_i$ and $|D| = \sum_{i=m+1}^{m+n} a_i$. When $|S| = |D|$, the logistic stage is balanced, otherwise ($|S| > |D|$ or $|S| < |D|$) and the logistic stage is unbalanced.

The same kind of logistic nodes make up of a logistic interface, such as if s is the supplier interface and d is the customer interface. The logistic distribution on each interface is named as its interface distribution.

In a balanced multi-stage logistic system, the interface distribution on each interface is unique. In other words, the interface distribution is a constant. As a result, the layout procedure on each stage is independent each other.

However, in an unbalanced multi-stage logistic system, the interface distribution is not unique, but a parameterized variable. Consequently, the layout procedure on each stage is dependent on each other stage. If the logistic distribution on each interface is already known, the layout procedure on each stage is independent each other, just as the balanced one. That is, the parameterized interface distribution can convert an unbalanced multi-stage logistic system to a set of balanced ones.

With the chromosome, which consists of the logis-

Received 2004-09-17.

Biographies: Xu Hang (1975—), male, graduate; Ye Qingtai (corresponding author), male, professor, yqingtai@online.sh.cn.

tic distributions on all interfaces, the genetic algorithm can search the whole solution space.

However, the layout procedure on each stage is a linear programming. When there are $l - 1$ stages in the logistic system, and the population size is o , evolution generation is g , the GA will carry $(l - 1)og$ times linear programming, which needs a large amount of computational time.

If the logistic pattern on each stage can be encoded and the coding is the sub-string of the chromosome, the layout procedure of the logistic pattern on each stage can be solved by GA.

2 Prüfer Number and Improved Decoding Procedure

Cayley proved that for a complete graph with p nodes, there are $p^{(p-2)}$ distinct labeled trees. Prüfer presented the simplest proof of Cayley's formula by establishing a one-to-one correspondence between the set of spanning trees and a set of $p - 2$ digits with an integer between 1 and p inclusive^[1,2]. The verification for the excellence of the Prüfer number has been addressed in Refs. [5, 6].

2.1 Shortcoming of Syarif's decoding procedure

For example, a transportation network of a one-stage logistic system is shown in Fig. 1. 1, 2, 3 and 4 are the supplier nodes, and 5, 6 and 7 are the demand nodes. The pair of numbers beside every node designates the maximal and residual capacity of the node. In Syarif's method^[1], the capacities of the nodes of the logistic network are not considered. As a result, the corresponding spanning tree of a Prüfer number $P = [4, 2, 6, 6, 6]$ is shown in Fig. 1.

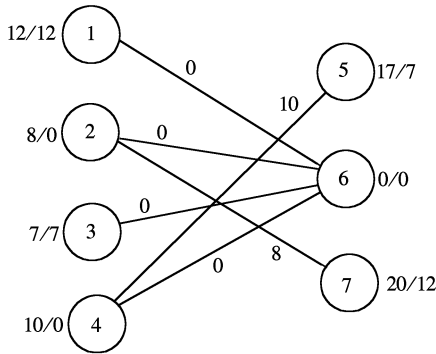


Fig. 1 Spanning tree gotten from $P = [4, 2, 6, 6, 6]$ by Syarif's method

In Fig. 1, the residual capacities of nodes 1, 3, 5 and 7 are not zero. That is, among both suppliers and customers, some nodes have residual capacities. In other words, this spanning tree cannot satisfy the de-

mand of this logistic stage. We name this phenomenon as non-matching between the spanning tree and the logistic stage, and name the spanning tree as a non-matching tree of the logistic stage. Although some additional repairing rules are used to amend Syarif's method, the computational efficiency will greatly be weakened.

2.2 The improved decoding procedure

To overcome the shortcoming of Syarif's decoding procedure, an improved decoding procedure is put forward.

Nomenclature If P is an ordinal number list, the leftmost digit of P is marked as \bar{P} . Practically, a Prüfer number is an ordinal number list. \bar{P} is the ordinal number list of all digits not included in P . \bar{P} is sorted in ascending order.

Suppose $i \in s \cup d$, all the digits among P , which are in the same set (s or d) with i , are marked as $P(i)$, we define that $i \notin P(i)$. The leftmost digit of $P(i)$ is marked as $\bar{P}(i)$; all the digits among P , which are not in the same set (s or d) with i , are marked as $P(\hat{i})$, and the leftmost digit of $P(\hat{i})$ is marked as $\bar{P}(\hat{i})$; all the digits among \bar{P} , which are in the same set (s or d) with i , are marked as $\bar{P}(i)$, we define that $i \notin \bar{P}(i)$. The leftmost digit of $\bar{P}(i)$ is marked as $\bar{\bar{P}}(i)$.

Step 1 Let P be the original Prüfer number produced stochastically, let R_i denote the number of appearances of nodes i in the Prüfer number P and L_i denote the number of connections of the nodes i for all $i \in s \cup d$, $L_i = R_i + 1$, if $\sum_{i=1}^m L_i = \sum_{i=m+1}^{m+n} L_i$, go to step 3; otherwise go to step 2.

Step 2 Repeat this step, until $\sum_{i=1}^m L_i = \sum_{i=m+1}^{m+n} L_i$. If $\sum_{i=1}^m L_i > \sum_{i=m+1}^{m+n} L_i$, then select one digit i in P ($i \in s$), and replace it with the number j ($j \in d$); otherwise, select one digit i in P ($i \in d$), and replace it with the number j ($j \in s$).

Step 3 According to the modified P , \bar{P} is constructed, which is the ordinal number list of all nodes not included in P . \bar{P} indicates eligible nodes for consideration in building a tree. \bar{P} is sorted in ascending order.

Step 4 Repeat the following process until no digits are left in P . ① Let i be the smallest label node in \bar{P} . Let j be the leftmost digit of $P(\hat{i})$ which is not in the same set (s or d) with i . That is $j = \bar{P}(\hat{i})$. ② Add the edge from i to j into the tree. Assign $x_{ij} =$

$\min\{a_i, a_j\}$. Update the availability $a_i = a_i - x_{ij}$ and $a_j = a_j - x_{ij}$. ③ If $a_i = 0$, remove j from P and i from \bar{P} . If j does not appear anywhere in the remaining part of P , put it into \bar{P} . Designate i as no longer eligible.

Otherwise, if $a_j = 0$, keep i in \bar{P} and delete $\bar{P}(i)$ from P . If the deleted digit $\bar{P}(i)$ does not appear in P anymore, add it to \bar{P} . If $P(j) \neq \emptyset$, replace every digit j in P with $\bar{P}(j)$; otherwise, replace every digit j in P with $\bar{P}(j)$, and delete $\bar{P}(j)$ from \bar{P} .

Step 5 If no digit remains in P , there are exactly two nodes, r and s , still eligible for consideration, and $a_r = a_s$. Add the edge from r to s into the tree and form a tree with $m + n - 1$ links. $x_{rs} = \min\{a_r, a_s\} = a_r = a_s$, $a_s = a_s - x_{rs} = 0$, $a_r = a_r - x_{rs} = 0$; at this time, because $\sum_{i=1}^m a_i = \sum_{i=m+1}^{m+n} a_i$, $\forall i \in s \cup d, a_i = 0$.

By the improved decoding procedure, the Prüfer number $P = [4, 2, 6, 6, 6]$ can be decoded to the logistic pattern, as shown in Fig. 2.

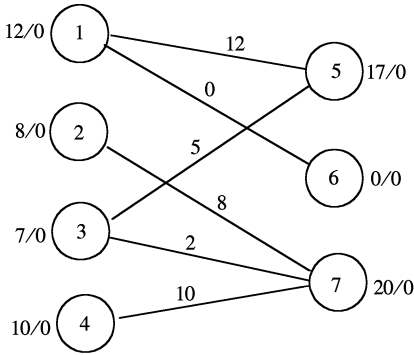


Fig. 2 Logistic pattern of $P = [4, 2, 6, 6, 6]$ by the improved method

3 Improved Decoding Procedure in Unbalanced Stage

A Prüfer number can just encode a spanning tree without the weight on the edge. In a balanced logistic stage, there is just a single feasible logistic pattern on a single spanning tree and the weight on each edge is unique, so a Prüfer number is enough to represent a logistic pattern.

However, in an unbalanced one, there are perhaps several feasible logistic patterns on a single spanning tree. A Prüfer number cannot provide sufficient information to represent a pattern in an unbalanced logistic stage. For example, in the unbalanced logistic stage ($|S| > |D|$) shown in Fig. 3, the two logistic patterns have the same spanning tree, but the weights on the edge $(1, 5)$, $(3, 5)$, $(2, 7)$, $(4, 7)$ are different.

As a result, in a balanced logistic stage, for any logistic pattern, there is always at least a Prüfer num-

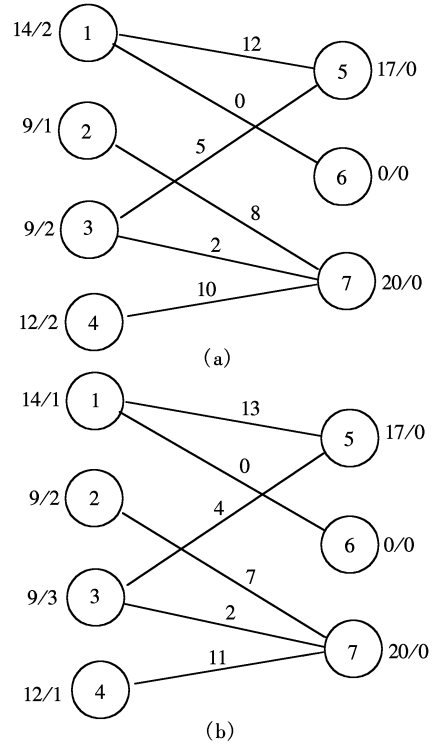


Fig. 3 Two different logistic patterns on the same spanning tree in an unbalanced logistic stage

ber, which can be decoded to it. However, in an unbalanced logistic stage, some logistic patterns will never be gotten by decoding any Prüfer numbers, no matter the improved method or Syarif's.

So, the straightforward method to solve this problem is to convert an unbalanced logistic system to a balanced one, which is fitly the business of the parameterized interface distribution. In other words, if the parameterized interface distribution is removed from the chromosome, the genetic algorithm with the Prüfer number coding cannot search the whole solution space of an unbalanced logistic system. Consequently, the parameterized interface distribution is the absolutely necessary ingredient of the chromosome.

By the interface distribution coding, the selection of the middle facilities can also be encoded. When the interface distribution of a node is 0, this node is not selected and should not be open. Then the 0-1 coding used by Syarif et al. [11] is not necessary.

4 Numerical Example

From the above researches, a new genetic algorithm based on parameterized interface distribution and Prüfer number coding (PIP-GA) comes into being [7,8]. A test problem, which is given in Tab. 1 to Tab. 4, is to find a feasible logistic pattern of the unbalanced multi-stage logistic system with the minimum cost. To compare the effectiveness of PIP-GA,

this test problem is also solved by st-GA. In Fig. 4, the best logistic pattern given by PIP-GA is shown, which is never gotten by the st-GA in theory.

Tab. 1 Capacity and fixed cost for the test problem

Supplier	Capacity			Fixed cost	
	Plant	DC	Customer	Plant	DC
260	350	500	180	1 800	1 000
300	450	550	280	1 000	1 100
450	600	400	160	1 100	1 900
360	550	600	340	2 100	1 000
480	500	450	240	1 200	1 800

Tab. 2 The unit shipping cost from suppliers to plants

Suppliers	Plants				
	1	2	3	4	5
1	5	2	7	3	4
2	3	6	2	7	7
3	2	8	7	2	1
4	4	7	8	6	9
5	3	5	3	4	6

Tab. 3 The unit shipping cost from Plants to DCs

Plants	DCs				
	1	2	3	4	5
1	9	4	6	5	4
2	3	3	5	7	6
3	2	7	9	4	8
4	6	5	7	6	4
5	4	3	4	2	5

Tab. 4 The unit shipping cost from DCs to customers

DCs	Customers				
	1	2	3	4	5
1	2	4	7	5	6
2	5	3	2	5	3
3	6	5	4	4	2
4	4	5	3	3	2
5	3	4	5	6	5

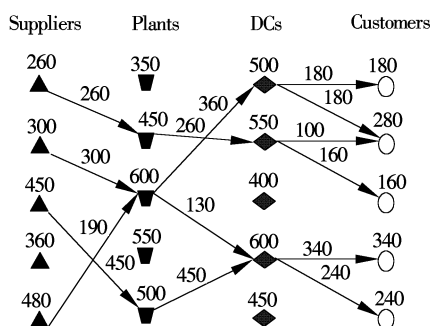


Fig. 4 Illustration of the best logistic pattern from PIP-GA method

With the higher decoding efficiency, PIP-GA requires less computational time on the decoding of the Prüfer number than st-GA. Although the computational time of the genetic operation on the interface distribution coding is just a little more than that on the 0-1 coding, the overall computational time of PIP-GA is

still less than that of st-GA.

However, because the interface distribution coding requires more memory than the 0-1 coding, the memory required for computation of PIP-GA is a little larger than that of st-GA.

After all, although requiring a little more consumption of memory, PIP-GA still has better results, not only on the best solution but also on the average solution, and requires less time consumption than st-GA, as shown in Tab. 5.

Tab. 5 Comparison of PIP-GA and st-GA

Method	Logistic cost			Computational time/s	Memory/Byte
	Best	Worst	Mean		
PIP-GA	14 660	14 660	14 660	15.8	39
st-GA	15 110	16 180	15 324	18.6	34

Note: Memory is used by a chromosome.

5 Conclusion

Based on the parameterized interface distribution and the improved decoding procedure of the Prüfer number, a new modeling method of the unbalanced multi-stage logistic system comes into being. The corresponding genetic algorithm, PIP-GA, not only can find better solution, but also requires less computational time than st-GA.

Although requiring a little more consumption of memory, we still believe PIP-GA is an efficient and robust method in the modeling and optimization of unbalanced multi-stage logistic systems.

References

- [1] Syarif A, Yun Y S, Gen M. Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach [J]. *Computers and Industrial Engineering*, 2002, **43**(1, 2): 299 – 314.
- [2] Syarif A, Gen M. Hybrid genetic algorithm for production/distribution system in supply chain [J]. *International Journal of Smart Engineering System Design*, 2003, **5**(4): 289 – 298.
- [3] Gen M, Cheng R. *Genetic algorithms and engineering optimization* [M]. New York: Wiley, 2000.
- [4] Gen M, Cheng R, Oren S. Network design techniques using adapted genetic algorithms [J]. *Advances in Engineering Software*, 2001, **32**(9): 731 – 744.
- [5] Zhou G, Min H, Gen M. The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach [J]. *Computers and Industrial Engineering*, 2002, **43**(1, 2): 251 – 261.
- [6] Zhou G, Zhu Y, Weng X, et al. The generalized approaches of genetic algorithms on constrained minimum spanning tree problems [A]. In: *Proc of World Congr Intelligent Control Autom*[C]. Piscataway: Institute of Electrical

and Electronics Engineers Inc, 2004. 2141 – 2145.

[7] Syarif A, Gen M. Solving exclusionary side constrained transportation problem by using a hybrid spanning tree-based genetic algorithm [J]. *Journal of Intelligent Manu-*

facturing, 2003, **14**(3, 4): 389 – 399.

[8] Zhou G, Gen M. A genetic algorithm approach on tree-like telecommunication network design problem [J]. *Journal of the Operational Research Society*, 2003, **54**(3): 248 – 254.

非平衡多阶段物流系统建模与优化方法

徐 杭 徐 榕 叶庆泰

(上海交通大学机械与动力工程学院, 上海 200030)

摘要: 首先提出了参数化界面分布的新概念, 将一个非平衡多阶段物流系统转化为多个相互独立的单一阶段物流系统, 然后采用 Prüfer 数对每个阶段上的物流模式进行编码. 通过一种改进的解码方法, 任何一个随机产生的 Prüfer 数都能够被解码为一个与物流系统的节点容量相匹配的可行的物流模式. 基于这 2 点创新, 建立了一种新的基于参数化界面分布和 Prüfer 数编码的系统建模方法, 相应的遗传算法称为 PIP-GA 方法. 与 st-GA 方法相比, PIP-GA 不但具有更好的优化结果, 而且需要的计算时间更少; 虽然所需的存储空间有所增加, PIP-GA 仍然是一种有效而稳健的非平衡多阶段物流系统建模与优化方法.

关键词: 物流; 优化; 遗传算法; Prüfer 数; 生成树; 参数化界面分布; 非平衡多阶段物流系统

中图分类号: TP18