

文章编号: 1001-0920(2013)10-1536-05

基于混合概率解发掘算法的第四方物流弹性网络设计

李锐, 黄敏, 王兴伟

(东北大学 a. 信息科学与工程学院, b. 流程工业综合自动化国家重点实验室, 沈阳 110819)

摘要: 为了使第四方物流系统能够安全、有效地运作, 研究基于弹性的第四方物流网络设计问题. 建立以网络构建成本为约束、弹性为目标的优化模型, 针对问题模型设计基于最优计算预算分配的混合概率解发掘算法. 通过随机产生的不同规模问题, 对混合概率解发掘算法进行性能评估, 并将实验结果与传统的概率解发掘算法进行比较. 实验结果表明了模型与算法的有效性.

关键词: 第四方物流; 网络设计; 弹性; 概率解发掘算法; 最优计算预算分配

中图分类号: TP273

文献标志码: A

Resilient network design for fourth-party logistics based on hybrid probability solution discovery algorithm

LI Rui, HUANG Min, WANG Xing-wei

(a. College of Information Science and Engineering, b. State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China. Correspondent: HUANG Min, E-mail: mhuang@mail.neu.edu.cn)

Abstract: In order to make the fourth party logistics(4PL) system operate safely and efficiently, the 4PL network design problem based on resilience is studied. An optimization model is developed, which maximizes the resilience of the network subjective to investment costs. A hybrid probability solution discovery algorithm(PSDA) based on optimal computing budget allocation(OCBA) is introduced. Finally, the performance of the hybrid algorithm is evaluated on randomly generated instances of different sizes, and the results are compared to those obtained by the traditional PSDA. The experimental results show the effectiveness of the proposed algorithm.

Key words: fourth party logistics; network design; resilience; probability solution discovery algorithm; optimal computing budget allocation

0 引言

第四方物流(4PL)^[1]专门为第一方、第二方和第三方提供物流规划、咨询和管理, 它依靠优秀的第三方物流(3PL)供应商、技术供应商、管理咨询和其他增值服务商, 帮助企业有效整合资源和提供广泛的供应链解决方案. 4PL网络设计的主要问题是确定产品从供应点到需求点的结构, 即通过选择第三方物流服务节点和运输供应商构建4PL网络.

弹性工程^[2-3]作为一个新的热点问题, 越来越受到人们的关注, 它是研究复杂系统安全性和稳定性

的一种有效方法. 一些学者已经对网络弹性的定义和分析问题进行了研究^[4-6]; 同时, 有关物流或供应链系统的安全性问题也得到学者的关注和研究^[7-10].

本文研究基于弹性的4PL网络优化问题, 建立了考虑随机中断的优化模型, 设计混合概率发掘算法(PSDA)^[11], 并通过数值例子与传统的PSDA比较. 仿真实验结果表明了混合PSDA方法的有效性.

1 问题描述

4PL网络可以表示为如图1所示的多重图, 图1中节点表示供应节点、3PL服务节点和需求节点, 弧

收稿日期: 2012-08-13; 修回日期: 2013-03-03.

基金项目: 国家自然科学基金项目(71071028, 70931001, 71021061, 61070162); 国家杰出青年科学基金项目(61225012); 高等学校博士学科点专项科研基金优先发展领域课题(20120042130003); 高等学校博士学科点专项科研基金项目(20110042110024, 20100042110025); 工信部物联网发展专项资金项目; 中央高校基本科研业务费专项基金项目(N110204003).

作者简介: 李锐(1985-), 男, 博士生, 从事物流优化、智能算法的研究; 黄敏(1968-), 女, 教授, 博士生导师, 从事供应链管理、物流优化、智能算法等研究.

表示 3PL 运输供应商. 因为两个节点之间可能存在多个 3PL 供应商, 所以可能有多条弧. 3PL 服务节点或运输供应商以一定的概率随机中断, 从而导致运输成本提高甚至供应中断. 弹复性的网络要有抗击中断的能力, 使系统在中断状态下仍能够以较低的成本为客户提供满意服务.

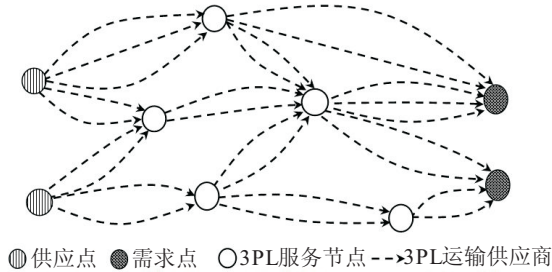


图 1 4PL 网络结构

2 问题建模

2.1 参数定义

已知 4PL 网络, $V = L \cup S \cup U$ 为节点集合, L 为需求节点的集合, S 为供应节点的集合, U 为 3PL 服务节点的集合; K_{ij} 为节点 i 和 j 之间 3PL 运输供应商的集合; F_s 为供应节点 $s \in S$ 的供应量; D_l 为需求节点 $l \in L$ 的需求量; $CE_{ijk}, QE_{ijk}, PE_{ijk}, HE_{ijk}$ 分别为节点 $i \in V$ 和 $j \in V$ 之间 3PL 运输供应商 $k \in K_{ij}$ 的单位货物的运输成本、运输能力、故障概率和固定运营费用; CN_u, QN_u, PN_u, HN_u 分别为 3PL 服务节点 $u \in U$ 的单位货物的处理成本、处理能力、故障概率和固定运营费用.

2.2 决策变量

令 $x_{ijk} \in \{0, 1\}$, 如果点 i 与 j 之间的 3PL 运输供应商 k 被选择, 则 x_{ijk} 为 1, 否则为 0; $y_u \in \{0, 1\}$, 如果 3PL 服务节点 u 被选择, 则 y_u 为 1, 否则为 0. 令 $X = \{x_{ijk} | \forall i \in V, \forall j \in V, \forall k \in K_{ij}\}$, $Y = \{y_u | \forall u \in U\}$.

2.3 问题的模型

根据以上参数和变量的定义, 基于弹复性的 4PL 网络设计模型如下:

$$\max \text{Res}(X, Y). \tag{1}$$

$$\text{s.t.} \sum_{u \in U} HN_u y_u + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K_{ij}} HE_{ijk} x_{ijk} < C_0; \tag{2}$$

$$x_{ujk} \leq y_u, \forall u \in U, \forall j \in V, \forall k \in K_{uj}; \tag{3}$$

$$x_{juk} \leq y_u, \forall u \in U, \forall j \in V, \forall k \in K_{ju}; \tag{4}$$

$$\sum_{j \in V} \sum_{k \in K_{uj}} x_{ujk} \geq y_u, \forall u \in U; \tag{5}$$

$$\sum_{j \in V} \sum_{k \in K_{ju}} x_{juk} \geq y_u, \forall u \in U; \tag{6}$$

$$\sum_{j \in V} \sum_{k \in K_{jl}} x_{jlk} \geq 1, \forall l \in L; \tag{7}$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in K_{ij}; \tag{8}$$

$$y_u \in \{0, 1\}, \forall u \in U. \tag{9}$$

目标函数 (1) 最大化网络弹复性; 约束 (2) 确保总固定成本小于投入成本 C_0 ; 约束 (3) 和 (4) 确保当 3PL 服务节点没有被选择时, 与之相连的 3PL 运输供应商也不能被选择; 约束 (5) 和 (6) 表示如果 3PL 服务节点被选择, 则必须保证至少分别有 1 个 3PL 运输供应商运入商品和运出商品; 约束 (7) 表示需求点必须至少有 1 个 3PL 运输供应商运入商品; 约束 (8) 和 (9) 表示二值的决策变量.

2.4 4PL 网络弹性

4PL 网络的弹性定义为

$$\text{Res}(X, Y) = \sum_{\theta \in \Theta} P^\theta \left[\frac{C_{\max} - \Phi^\theta(X, Y)}{C_{\max} - C_{\min}} \right]. \tag{10}$$

其中: P^θ 为故障状态 θ 发生的概率; Θ 为故障状态集合; C_{\min} 为潜在网络的最小运作费用; $C_{\max} = PV \sum_{l \in L} D_l$ 为网络的最大运作费用, PV 为单位商品缺货的惩罚值并且等于潜在网络的单位商品的最大运输及处理成本; $\Phi^\theta(X, Y)$ 为故障情景 $\theta \in \Theta$ 下的可变的运作成本, 通过如下模型确定:

$$\begin{aligned} \Phi^\theta(X, Y) = \min & \sum_{i \in V} \sum_{j \in V} \sum_{k \in K_{ij}} CE_{ijk} f_{ijk}^\theta + \\ & \sum_{i \in V} \sum_{u \in U} \sum_{k \in K_{iu}} CN_u f_{iuk}^\theta + \\ & PV \left[\sum_{l \in L} D_l - \sum_{l \in L} \sum_{i \in V} \sum_{k \in K_{il}} f_{ilk}^\theta \right]. \end{aligned} \tag{11}$$

$$\text{s.t.} f_{ijk}^\theta \leq (1 - a_{ijk}^\theta) x_{ijk} QE_{ijk}, \tag{12}$$

$$\forall i, j \in V, \forall k \in K_{ij};$$

$$\sum_{j \in V} \sum_{k \in K_{uj}} f_{ujk}^\theta \leq (1 - b_u^\theta) y_u QN_u, \forall u \in U; \tag{13}$$

$$\sum_{j \in V} \sum_{k \in K_{sj}} f_{sjk}^\theta \leq F_s, \forall s \in S; \tag{14}$$

$$\sum_{i \in V} \sum_{k \in K_{il}} f_{ilk}^\theta \leq D_l, \forall l \in L; \tag{15}$$

$$\sum_{j \in V} \sum_{k \in K_{uj}} f_{ujk}^\theta = \sum_{j \in V} \sum_{k \in K_{ju}} f_{juk}^\theta, \forall u \in U; \tag{16}$$

$$f_{ijk}^\theta \geq 0, \forall i, j \in V, \forall k \in K_{ij}. \tag{17}$$

目标函数 (11) 最小化网络的运作费用, 包括运输费用、处理费用和缺货惩罚费用, f_{ijk}^θ 为决策变量, 表示故障 θ 下节点 i 与 j 之间 3PL 运输供应商 k 的运输量; 约束 (12) 为 3PL 运输供应商能力的限制, a_{ijk}^θ 表示故障 θ 下点 i 与 j 之间的运输供应商 k 是否中断, $a_{ijk}^\theta = 1$ 为中断, $a_{ijk}^\theta = 0$ 为正常; 式 (13) 为 3PL 服务节点处理能力的约束, b_u^θ 表示故障 θ 下服务节点 u 是否中断, $b_u^\theta = 1$ 为中断, $b_u^\theta = 0$ 为正常; 约束 (14) 表示

供应点的最大供应量; 约束 (15) 表示对需求点的供应量不超过需求量; 约束 (16) 保证 3PL 服务节点两端的流平衡; 约束 (17) 为流量的非负条件.

3 算法设计

第四方物流弹性网络设计问题是经典网络可靠性设计问题的扩展, 因此也是 NP-hard 的. 对于大规模问题很难用精确算法求解, 所以设计一种混合概率解发掘算法. PSDA 算法是由 Ramirez-Marquez 等提出的一种进化算法, 可以成功求解网络可靠性问题^[11]和网络阻断问题^[12]. 本文引入最优计算预算分配方法 (OCBA)^[13]作为 PSDA 解的分析方法, 保证正确选择的最好解集合的概率最大, 从而提高算法的性能.

3.1 解的表示

如表 1 所示, 一个解可以表示为一维的二进制数组, 二进制码表示对应的 3PL 服务节点和运输供应商的选择情况, “1”为选择, “0”为未选择. 例如, (1,3) 对应的二进制数为“110”, 表示点 1 和点 3 之间的 3 个运输供应商中的第 2 个和第 3 个被选择.

表 1 解的表示

节点对 二进制数	3PL 服务节点		3PL 运输供应商		
	(2,2)	(3,3)	(1,2)	(1,3)	(3,4)
	1	1	011	110	010

3.2 样本解的产生

根据概率向量 γ_u , 利用 Monte Carlo 仿真产生数量为 SAMPLE 的解集合, $X_u^h (h = 1, 2, \dots, \text{SAMPLE})$ 的第 j 位由下式产生:

$$x_{uj}^h = \begin{cases} 1, & \text{rand}(0, 1) < \gamma_{uj}; \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

其中 γ_{uj} 为第 u 次循环第 j 位取值为 1 的概率.

由于解必须满足约束 (3)~(7), 按广度优先搜索依次确定每位的值. 如果一个 3PL 服务节点 $u \in U$ 有入弧无出弧, 则随机选择一个以 u 为起点的弧, 并将解中该弧的对应位值设为 1, 即至少选择 1 个出弧. 如果一个解中没有弧指向某需求节点 $l \in L$, 则可以选择类似的方法进行修复.

3.3 基于 OCBA 的解分析

首先计算根据概率产生的解 X_u^h 的适应值. 将构建成本约束 (2) 作为惩罚项加入适应值, 则 X_u^h 的适应值为

$$f(X_u^h) = \hat{\text{Res}}(X, Y) - \frac{(C(X, Y) - C_0)^+}{C_0}. \quad (19)$$

其中: $C(X, Y)$ 为构建成本; $\hat{\text{Res}}(X, Y)$ 为通过 Monte Carlo 仿真计算的弹性网络弹性 (详见第 3.4 节); $(\cdot)^+$ 号内值为正则取该值, 否则取零.

每一代根据适应值对 $f(X_u^h)$ 降序排列, 为了使得在解探索步骤中选择 m 个最好解子集 S_m 的概率最大, 利用 OCBA 方法分配仿真次数, 选择最好的子集.

定理 1 假设 k 系统的指标是随机变量, 均值和方差分别为 J_1, J_2, \dots, J_k 和 $\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2$. 给定总的仿真样本数量 T , 将其分配给各个系统, 当 $T \rightarrow \infty$ 时, 正确选择 m 个最好解的近似概率能够渐进最大化, 即

$$\frac{N_i}{N_j} = \left(\frac{\sigma_i/\delta_i}{\sigma_j/\delta_j} \right)^2, \quad i, j \in \{1, 2, \dots, k\}, i \neq j. \quad (20)$$

其中: N_i 为分配给 i 的样本数量; $\delta_i = c - \bar{J}_i$, $c = (\bar{J}_m + \bar{J}_{m+1})/2$, \bar{J}_i 为样本均值; σ_i 由样本方差近似.

根据定理 1, 基于 OCBA 的解分析步骤如下.

Step 1: 对于所有系统执行 n_0 次仿真, 计算样本均值 \bar{J}_h 和方差 s_h^2 , 初始化 $l \leftarrow 0$, $N_1^l = N_2^l = \dots = N_{\text{SAMPLE}}^l = n_0$.

Step 2: 若 $\sum_{h=1}^k N_h^l \geq T$, 则终止, 否则转至 Step 3.

Step 3: 增加仿真次数 Δ , 根据定理 1 计算新的仿真次数分配 $N_1^{l+1}, N_2^{l+1}, \dots, N_{\text{SAMPLE}}^{l+1}$.

Step 4: 对于每个设计 h 再执行 $\max[0, N_h^{l+1} - N_h^l]$ 次仿真, 令 $l \leftarrow l + 1$, 转至 Step 2.

Step 5: 对于 $h = 1, 2, \dots, \text{SAMPLE}$, 计算 X_u^h 的适应值 $f(X_u^h)$, 并按降序排列, 有

$$f(X_u^1) \geq f(X_u^2) \geq \dots \geq f(X_u^{\text{SAMPLE}}). \quad (21)$$

3.4 弹性计算

基于 Monte Carlo 仿真计算弹性步骤如下.

Step 1: 根据 3PL 服务节点和 3PL 运输供应商的故障概率, 随机生成产生 N 个故障状态.

Step 2: 对于每一种故障状态, 通过增加虚的起始节点和终止节点, 将问题转换为网络的最小费用最大流问题, 然后利用 Bellman-Ford 和 Ford-Fulkerson 相结合的方法求解.

Step 3: 评估弹性期望值

$$\hat{\text{Res}}(X, Y) = \frac{1}{N} \sum_{\theta=1}^N \left[\frac{C_{\max} - \Phi^\theta(X, Y)}{C_{\max} - C_{\min}} \right]. \quad (22)$$

3.5 解探索

从已排序的数量为 SAMPLE 的解集合中, 选择数量为 m 的最好子集更新概率向量 γ_u , 同时选择数量为 TOP 的最好解存储于集合 K , 具体如下:

$$K = K \cup \{f(X_u^1), f(X_u^2), \dots, f(X_u^{\text{TOP}})\};$$

$$u = u + 1.$$

根据下式更新概率向量 γ_u :

$$\gamma_{uj} = \sum_{h=1}^m x_{(u-1)j}^h / m, \quad m \ll \text{SAMPLE}. \quad (23)$$

3.6 停止准则

当算法达到最大循环代数 NG, 或者 γ_u 不能再更新时, 算法终止.

3.7 HPSDA 步骤

Step 1: 初始化概率向量 γ_0 、最优子集规模 m 、解集规模 SAMPLE、最大循环代数 NG.

Step 2: 根据概率向量 γ_u , 基于广度优先搜索和修复策略, 利用 Monte Carlo 仿真产生 SAMPLE 个解.

Step 3: 将基于 OCBA 分析产生的解按降序排列.

Step 4: 选择数量为 TOP 的最好解存储于集合 K , 同时选择规模为 m 的最好子集更新概率向量 γ_u .

Step 5: 如果满足停止准则, 则终止循环, 否则转至 Step 2.

Step 6: 对集合 K 中的解执行大样本仿真重新评估弹性, 选择目标值最大的解输出.

4 仿真实验

为了测试 HPSDA 算法的有效性, 本文对随机产生的不同规模问题进行测试. 算法采用 Matlab 语言编码, 实验环境为 Intel Core 2 Quad 2.66 GHz 台式机, 内存 2.00 GB.

问题的参数按照均匀分布随机产生: 供应节点的供应量 $F_s \sim U[90, 160]$; 需求节点的需求量 $D_l \sim U[40, 70]$, 3PL 运输供应商的故障概率 $PE_{ijk} \sim U[0, 1]$, 单位货物运输成本 $CE_{ijk} \sim U[10, 50]$, 运输能力 $QE_{ijk} \sim U[100, 200]$, 构建成本 $HE_{ijk} \sim U[500, 1000]$; 3PL 服务节点的故障概率 $PN_u \sim U[0, 1]$, 单位货物处理成本 $CN_u \sim U[10, 50]$, 处理能力 $QN_u \sim U[100, 200]$, 构建成本 $HN_u \sim U[500, 1000]$.

表 2 给出每个问题两种算法 20 次运行求出的最优解 X^* 弹性估计值 $Res(X^*)$ 的最好值、最差值、平均值、标准方差和平均 CPU 时间的比较, 其中 NS, NU, NL, NE 分别表示供应点、3PL 服务节点、需求点和 3PL 运输供应商的数量. HPSDA 的参数设置为: $T = 2000, n_0 = 10, \Delta = 20$; 初始概率向量 $\gamma_{0j} = 0.5$, 最大循环代数 $NG = 35$, 最优子集规模 $m = 30$, 每代解集规模 $SAMPLE = 100$. 由表 2 可见, HPSDA 的求

解质量和稳定性均明显优于 PSDA, 由于 OCBA 过程的应用, HPSDA 的 CPU 时间略大于 PSDA.

为了分析 HPSDA 参数对算法性能的影响, 以问题 P2 为例进行实验, 并将算法运行 20 次的平均值作为性能指标. 影响算法性能的主要参数包括总仿真次数 T , 最优子集规模 m 和解集规模 SAMPLE. 图 2 给出总仿真次数 T 对算法性能的影响. 由图 2 可见, 随着 T 的增加, 平均值变大; 当 $T = 1800$ 时, 增加仿真次数平均值没有明显的变化. 由此可知, 应用 OCBA 并不需要太大的仿真次数即可得到满意的结果.

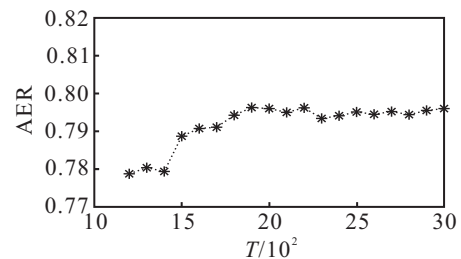


图 2 仿真次数对算法性能的影响

图 3 给出最优子集规模 m 对算法性能的影响. 其中: 解集规模 $SAMPLE = 80$, 最优子集规模 m 为 5 ~ 60. 由图 3 可见, 当 $m = 30$ 时平均值最大, m 的取值太大或者太小都会使算法的性能变差. m 变小使算法收敛速度加快, 但是容易陷入局部最优; 而 m 变大, 算法不容易收敛, 解的质量变差. 对于本文研究的问题, $m/SAMPLE$ 的取值范围应在 (0.2 ~ 0.5) 区间内.

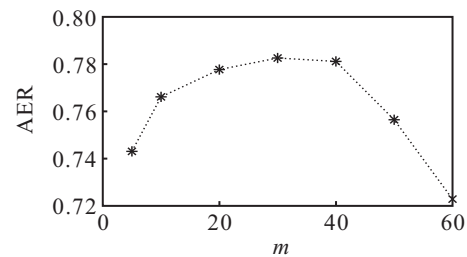


图 3 最优子集规模对算法性能的影响

图 4 为解集规模 SAMPLE 从 20 变化到 120 时, HPSDA 的性能变化情况. 由图 4 可见, 随着解集规模的变大, 算法性能提高, 当规模达到 100 时, 平均值

表 2 不同规模问题算法对比

问题	规模(NS*NU*NL*NE)	算法	最好值	最差值	平均值	标准方差	平均 CPU 时间/s
P1	2*5*2*25	HPSDA	0.746 2	0.730 3	0.735 3	0.006 3	123.557 9
		PSDA	0.706 3	0.688 3	0.698 1	0.007 6	112.693 9
P2	3*7*3*60	HPSDA	0.812 1	0.773 5	0.795 9	0.010 2	297.512 6
		PSDA	0.780 2	0.750 6	0.763 2	0.012 4	286.388 5
P3	4*9*4*121	HPSDA	0.850 2	0.828 5	0.839 9	0.007 5	522.060 6
		PSDA	0.829 1	0.808 7	0.815 2	0.008 4	532.739 5
P4	5*11*5*168	HPSDA	0.915 4	0.912 4	0.914 2	0.001 6	1406.400 0
		PSDA	0.901 4	0.894 7	0.897 0	0.003 8	1393.500 0

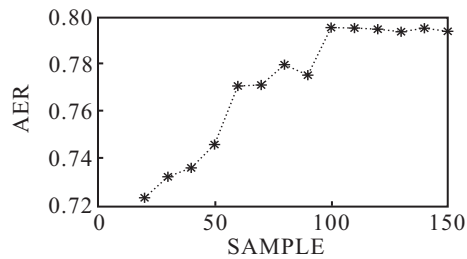


图4 解集规模对算法性能的影响

在0.79上下趋于平稳波动。

5 结论

本文研究基于弹复性的4PL网络设计问题,使4PL系统能够安全、有效地运行.给出基于弹复性的4PL网络设计模型,并针对问题求解的复杂性设计了HPSDA算法.通过随机数值例子验证了模型和算法的有效性,并分析了主要参数对算法性能的影响.

参考文献(References)

- [1] Gattorna J. Strategic supply chain alignment: Best practice in supply chain management[M]. Hampshire: Ashgate Pub Co, 1998: 425-445.
- [2] Hollnagel E, Woods D D, Leveson N. Resilience engineering: Concepts and precepts[M]. Aldershot: Ashgate, 2006: 35-41.
- [3] Madni A M, Jackson S. Towards a conceptual framework for resilience engineering[J]. IEEE System J, 2009, 3(2): 181-191.
- [4] Whitson J C, Ramirez-Marquez J E. Resiliency as a component importance measure in network reliability[J]. Reliability Engineering and System Safety, 2009, 94(10): 1685-1693.
- [5] Wang D W, Ip W H. Evaluation and analysis of logistics network resilience with application to aircraft servicing[J]. IEEE System J, 2008, 3(2): 166-173.
- [6] Konak A, Bartolacci M R. Designing survivable resilient networks: A stochastic hybrid genetic algorithm approach[J]. J of Management Science, 2007, 35(6): 645-658.
- [7] Rice J, Caniato F. Building a secure and resilient supply network[J]. Supply Chain Management Review, 2003, 7(5): 22-30.
- [8] Klibi W, Martel A, Guitouni A. The design of robust value-creating supply chain networks: A critical review[J]. European J of Operational Research, 2010, 203(2): 283-29.
- [9] Meepetchdee Y, Shah N. Logistical network design with robustness and complexity considerations[J]. Int J of Physical Distribution & Logistics Management, 2007, 37(3): 201-222.
- [10] 秦绪伟,唐立新.面向干扰事件的可靠物流系统设计研究进展[J].控制与决策,2010,25(2):161-165.
(Qin X W, Tang L X. Reliable logistics system design research under disruptions: Review[J]. Control and Decision, 2010, 25(2): 161-165.)
- [11] Ramirez-Marquez J E, Rocco C M. All-terminal network reliability optimization via probabilistic solution discovery[J]. Reliability Engineering and System Safety, 2008, 93(11): 1689-1697.
- [12] Ramirez-Marquez J E, Rocco C M. Stochastic network interdiction optimization via capacitated network reliability modeling and probabilistic solution discovery[J]. Reliability Engineering and System Safety, 2009, 94(5): 913-921.
- [13] Chen C H, He D H, Fu M C, et al. Efficient simulation budget allocation for selecting an optimal subset[J]. Inform J on Computing, 2008, 20(4): 579-595.

(上接第1535页)

- [3] Athuraliya S, Low S H, Yin Q. REM: Active queue management[J]. IEEE Network, 2001, 15(3): 397-413.
- [4] Sun J S, Zukerman M. RaQ: A robust active queue management scheme based on rate and queue length[J]. Computer Communications, 2007, 30(8): 1731-1741.
- [5] Barrera I D, Arce G R, Bohacek S. Statistical approach for congestion control in gateway routers[J]. Computer Networks, 2011, 55(3): 572-582.
- [6] Kunniyur S, Srikant R. An analysis and design of an adaptive virtual queue(AVQ) algorithm for active queue management[J]. AQM Computer Communication Review, 2001, 31(4): 123-134.
- [7] Ramakrishnan K K, Floyd S, Black D L. The addition of explicit congestion notification (ECN) to IP[Z]. RFC 3168, Internet Engineering Task Force, 2001.
- [8] Wang P, Chen H, Yang X P, et al. Design and analysis of a model predictive controller for active queue management[J]. ISA Trans, 2012, 51(1): 120-131.
- [9] Marami B, Haeri M. Implementation of MPC as an AQM controller[J]. Computer Communications, 2010, 33(2): 227-239.
- [10] Bigdeli N, Haeri M. Predictive functional control for active queue management in congested TCP/IP networks[J]. ISA Transactions, 2009, 48(1): 107-121.