

基于改进蚁群算法的CVRP问题*

张海军¹,徐廷学¹,逄程¹,韩玉²

(1. 海军航空大学,山东 烟台 264001 2. 海军航空大学青岛校区,山东 青岛 266041)

摘要:为了更好地求解CVRP问题,对蚁群算法进行了一定程度的改进。改进后的算法更加科学地初始化蚂蚁的位置,使蚂蚁有更大可能性地寻找到最优路径。在搜索的过程中与禁忌搜索算法结合,添加新的参数负信息素来记忆已经访问过的客户。同时,使用局部信息素更新和全局信息素更新相结合的信息素更新方式,并且全局信息素更新添加了动态更新的新模式。使用2-opt搜索对结果进行进一步的探索,扩大搜索的范围,增加了得到最优解的概率。

关键词: CVRP 蚁群算法 禁忌搜索 2-opt

中图分类号: TP18

文献标识码: A

DOI: 10.3969/j.issn.1002-0640.2019.01.013

引用格式: 张海军,徐廷学,逄程,等.基于改进蚁群算法的CVRP问题[J].火力与指挥控制,2019,44(1):67-71.

Research on CVRP Problem Based on Improved Ant Colony Algorithm

ZHANG Hai-jun¹, XU Ting-xue¹, LU Cheng¹, HAN Yu²

(1. Naval Aviation University, Yantai 264001, China;

2. Naval Aviation University Qingdao Branch, Qingdao 266041, China)

Abstract: In order to solve the problem of CVRP better, the ant colony algorithm is improved to some extent. First, the improved algorithm more scientifically initializes the position of the ants, making the ants more likely to find the optimal path. Then, in the search process with the tabu search algorithm, add a new parameter pheromone to remember the customer has been visited. At the same time, the use of local pheromone updates and global pheromone updates are combined with pheromone updating methods, and the global pheromone update adds a new pattern of dynamic updates. Finally, using the 2-opt search to further explore the results, expand the scope of the search, increasing the probability of getting the optimal solution.

Key words: CVRP ant colony algorithm tabu search 2-opt

Citation format: ZHANG H J, XU T X, LU C, et al. Research on CVRP problem based on improved ant colony algorithm[J]. Fire Control & Command Control, 2019, 44(1): 67-71.

0 引言

配送车辆的路径问题(Vehicle Routing Problem, VRP)是研究物流配送的关键,VRP一直是物流研究领域中的一个具有重要理论意义和现实意义的问题^[1-2]。因此,在物流保障过程中,需要制定合理有效的配送路径,选择省时省力的方法,在完成保

障任务的前提下,尽最大努力满足客户对配送的需求。

配送车辆的路径问题(Vehicle Routing Problem, VRP)是Dantzig和Ramser^[3]在1959年提出来的。所谓VRP,一般指的是:调用多辆车辆,从车场出发,对客户组织适当的行车路线,有序地访问所有客户并且只访问一次,在满足所有设定的约束条件

收稿日期:2017-10-17

修回日期:2018-01-28

* 基金项目:国家自然科学基金(51605487),山东省自然科学基金(ZR2016FQ03),中国博士后科学基金资助项目(2016M592965)

作者简介:张海军(1989-),男,山东威海人,硕士研究生。研究方向:器材、仓储勤务技术。

下,力争实现所设定的目标。带容量约束的车辆路径问题(CVRP)是车辆路径问题(VRP)中的基本问题之一,广泛应用于资源配置和物流运输等方面。

求解 CVRP 的算法大致可以分为精确算法和启发式算法两类^[4-5]。精确算法(Accurately Arithmetic)在求解规模较小的路径优化问题时能够在可承受的空间与时间消耗下得到精确解。但是配送路径优化问题属于 NP 问题,实际求解过程中问题的规模较大,使用精确算法所消耗的空间和时间成本都是比较大的。而启发式算法(Heuristics)不管是求解小规模的问题还是大规模的问题,都能够在一定的范围和较短的时间内给出满意解或最优解。因此,目前相关领域的专家以及研究人员对于求解算法的主要研究方向是启发式算法,特别是对现代启发式算法的研究。蚁群算法作为现代启发式算法之一,自从被提出之后就受到了广泛的关注,该算法具有正反馈性、并行性和鲁棒性等优点,在解决任务分配、路径优化等问题时表现出了良好的性能,但同时蚁群算法也存在一定的缺陷,如在解决大规模的问题时,会出现运算时间长、收敛速度慢、容易陷入局部最优解等问题^[6]。本文在基本的蚁群算法基础上进行合理改进,提高了算法的性能,使算法的运算结果更加准确。首先,制定转移规则,更加科学地初始化车辆的位置,使得车辆有更大概率地走出最佳的路径;然后,在搜索的过程中与禁忌搜索算法结合,添加新的参数负信息素来记忆已经访问过的客户;同时,使用局部信息素更新和全局信息素更新相结合的信息素更新方式,并且,全局信息素更新添加了动态更新的新模式;最后,使用 2-opt 搜索^[7]对结果进行进一步的探索,扩大搜索的范围,增加了得到最优解的概率。通过仿真实验,用不同的算法对已知案例进行仿真,结果显示本文算法相对于其他算法,无论是结果的准确性还是运算时间的长短都表现出了更好的性能。

1 问题描述及数学建模

CVRP 是从同一车场出发的多辆车辆访问多个客户进行运送货物的问题。已知出发车场的位置以及需要访问的客户的位置,客户的数量以及需求数量,车辆的数量以及负载数量,对客户组织适当的行车路线,有序地访问所有客户并且只访问一次,在满足所有设定的约束条件下,力争实现所设定的目标(车辆所走的路径最短或者花费最少)。

将现实问题转化为数学模型,可以将 CVRP 描述为 m 辆车从车场 0 出发访问 n 位客户进行运送货物的问题。设 d_i 为客户 i 的需求数量, c_{ij} 为车辆从客户 i 到客户 j 的运输花费, b_k 为车辆 k 的负载数

量,而

$$x_{ij}^k = \begin{cases} 1, & \text{车辆 } k \text{ 服务客户 } i \text{ 后接着服务客户 } j \\ 0, & \text{其他} \end{cases} \quad (1)$$

求解 CVRP 是为了使车辆走出的总的路径最短^[8-9],从而令运输的成本 J 最小,即

$$J = \min \sum_{k=1}^m \sum_{\substack{i,j=0 \\ i \neq j}}^n c_{ij} x_{ij}^k \quad (2)$$

且应满足以下约束条件:

$$\sum_{j=1}^n x_{0j}^k = \sum_{j=1}^n x_{j0}^k = 1, k \in (1, 2, \dots, m) \quad (3)$$

$$\sum_{k=1}^m \sum_{\substack{j=0 \\ j \neq i}}^n x_{ij}^k = 1, i \in (1, 2, \dots, n) \quad (4)$$

$$\sum_{k=1}^m \sum_{\substack{i=0 \\ i \neq j}}^n x_{ij}^k = 1, j \in (1, 2, \dots, n) \quad (5)$$

$$\sum_{i=1}^n d_i \sum_{\substack{j=0 \\ j \neq i}}^n x_{ij}^k \leq b_k, k \in (1, 2, \dots, m) \quad (6)$$

式(3)表示所有车辆都从同一个车场出发,完成所担负的任务之后全部返回车场,每一辆车走出的路径都形成一个哈密尔顿巡回;式(4)和式(5)表示车辆要对所有客户进行服务,并且只能服务一次;式(6)表示每辆车服务客户时,自身的负载量不能低于被服务客户的总的需求量。

2 算法设计

2.1 蚁群算法求解 CVRP

使用蚁群算法对 CVRP 进行求解时,蚂蚁会在各路径上产生信息素,信息素 τ_{ij} 表示车辆 k 在对客户 i 服务完成之后,下一个客户 j 对车辆 k 产生的吸引程度。当车辆的负载量能够满足客户的需求时,车辆按照设定好的转移规则来选定客户。在选好所有客户形成回路之后,对各路径上的信息素实行更新策略,进而实现一次迭代。在迭代达到最大次数时,可以获得一个 Pareto 解集。跟其他的启发式算法相比,蚁群算法具有更高的机能,但在对 CVRP 的求解过程中还是会有一些不可避免的缺陷。

2.1.1 选取初始客户过程中的局限

在 CVRP 中,因为车辆都有一定的负载,访问多个客户要满足客户需求,而每辆车在访问客户时都会有货物量限制和访问先后的问题,所以求解 CVRP 受到选取初始客户的极大影响。运用蚁群算法求解 CVRP 时,初始化蚂蚁位置的方式有两种:1) 全部放置方式。将所有蚂蚁全部放置于每个节点,这样搜索的范围会很大,更容易得到全局最优的解,但同时也将造成大量时间的消耗,在处理较大规模的 CVRP 时,会令问题更加复杂。2) 随机放

置方式。将部分蚂蚁随机放置到部分节点,这样做可以节省时间,降低问题的复杂程度,但是也容易造成得到的结果只是局部最优的,令每次迭代得到的 Pareto 解差距过大。

2.1.2 信息素更新过程的局限性。

随着算法的不断运算,寻找着最优路径,而信息素在路径上不停地累积,当前已寻找到的最优路径上的信息素越来越多,这或许会造成以下的不良后果:1) 在当前已寻找到的最优路径上的信息素过多,造成算法的停滞。2) 即便寻找新的最优路径,这条路径上的信息素也远远没有之前得到的最优路径上的信息素多,几次迭代之后,这种现象也可能没有得到缓解。总之,在基本蚁群算法中,原有的信息素更新方式会导致信息素的分布不能很好地体现当前最优路径的变化,这会对算法的求解过程造成很大的影响。

2.2 算法改进

针对在求解 CVRP 时蚁群算法存在的缺陷,本文改进蚁群算法将提高其求解 CVRP 的性能。

2.2.1 转移概率的改进

为了降低算法的复杂程度,同时加强算法的精确程度,在算法选择初始客户的过程中,本文将蚁群算法和禁忌搜索进行融合,修正转移概率,以改进蚁群算法。将禁忌搜索融入蚁群算法主要是利用禁忌搜索长久的记忆能力来加强蚁群算法搜索的开拓性^[10-12]。本文结合禁忌搜索记忆之前访问过的节点,加入了负信息素 ϕ_{ij} 的概念:

$$\phi_{ij} = \gamma / \lambda_{ij}(t) \quad (7)$$

对于每个边 (i, j) , γ 是常量, $\lambda_{ij}(t)$ 是边 (i, j) 的应用频率(表示在得到的 Pareto 解中节点 i 和 j 被出现的次数)。每当边 (i, j) 或 (j, i) 被包含在任一解中, $\lambda_{ij}(t)$ 就增加 1; $\lambda_{ij}(0) = 1$ 。

当蚂蚁 k 处于节点 i 的位置时,根据以下规则选择下一节点:

$$j = \begin{cases} \arg \max \{ \tau_{iu}^\alpha(t) \eta_{ij}^\beta(t) \phi_{ij}^\gamma(t) \}, & \text{如果 } r \leq r_0 \\ J, & \text{如果 } r > r_0 \end{cases} \quad (8)$$

式中, $r \sim U(0, 1)$, $r_0 \in [0, 1]$ 由用户指定具体参数值; $J \in N_i^k(t)$ 是根据以下转移概率随机选择的节点:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{iu}^\alpha(t) \eta_{ij}^\beta(t) \phi_{ij}^\gamma(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t) \phi_{iu}^\gamma(t)}, & \text{如果 } (i, j) \in N_i^k(t) \\ 0, & \text{其他情况} \end{cases} \quad (9)$$

式中, τ_{ij} 表示边 (i, j) 上的信息素浓度, η_{ij} 表示从一个节点转移到另一个节点的吸引度,在这里可以设

$$\eta_{ij} = \frac{1}{d_{ij}}, \quad d_{ij} \text{ 表示节点 } i \text{ 与节点 } j \text{ 之间的距离}, N_i^k(t)$$

是可选节点的集合。基于式中的转移概率,边 (i, j) 被用于解中的次数越多,负信息素 ϕ_{ij} 的值就越小,所以和其他被用于解中次数少的边相比,边 (i, j) 上 ϕ_{ij} 的值也就越小,这有利于对被用于解中次数少的边的探索。而大的 γ 值会限制禁忌搜索的记忆效果和探索性,因而可以在初期使用小的 γ 值以利于搜索,并随着迭代次数的增加而增大 γ 值。

2.2.2 信息素更新方式

针对蚁群算法在信息素更新过程中存在的问题,本文使用局部信息素更新和全局信息素更新相结合的信息素更新方式,并且,全局信息素更新添加了动态更新的新模式。

1) 局部更新。在构建最优路径的过程中,蚂蚁每经过边 (i, j) 都会依照下面的信息素更新公式来更新这条边上的信息素:

$$\tau_{ij}(t+1) = (1 - \rho_1) \tau_{ij}(t) + \rho_1 \tau_0 \quad (10)$$

其中, ρ_1 表示局部更新的挥发系数, $\rho_1 \in (0, 1)$, τ_0 表示信息素的初始值,是一个较小的正常量,可以设

$$\tau_0 = \frac{1}{n_G L}, \quad n_G \text{ 表示节点数}, L \text{ 表示生成的路径长度。}$$

引入信息素局部更新的目的,是为了使蚂蚁在经过边 (i, j) 时,会挥发这条边上一部分的信息素,降低其他的蚂蚁同样选择这条边的几率,可以更多地去探索其他的边,从而令算法的开拓性大大加强。

2) 全局动态更新。当迭代一次完成后,按下式对此次迭代得到的最优路径上的信息素实行全局更新:

$$\tau_{ij}(t+1) = (1 - \rho) \tau_{ij}(t) + \rho \Delta \tau_{ij} \quad (11)$$

$$\Delta \tau_{ij} = \frac{L_2 - L_1}{L_2} \quad (12)$$

其中, ρ 表示全局更新的挥发系数, $\rho \in (0, 1)$, $\Delta \tau_{ij}$ 表示信息素的增加量, L_1 为 x 次迭代时得到的最优路径长度, L_2 为 $x-1$ 次迭代之后所有迭代中的最优路径长度。对最优路径上的信息素实行更新策略是为之后迭代过程中的寻优工作提供信息素的正反馈,使得在之后的迭代中能够更好更快地找到最优的路径。

式(11)和式(12)可以依照每次迭代得到的最优路径的情况,对当前迭代所得的最优路径上的全局信息素进行动态调整。在 x 次迭代后,得到了一个最优路径,相比 $x-1$ 次迭代,这条路径要短, L_2 和 L_1 的差值就为正,说明当前迭代找到的最优路径比之前寻找到的要好,但此时该路径上的信息素浓度并

不高,而根据式(12) L_2 和 L_1 的差值越大, $\Delta\tau_{ij}$ 就会越大,在之后的迭代中,信息素浓度的积累就会越快; L_2 和 L_1 的差值越小, $\Delta\tau_{ij}$ 就会越小,在之后的迭代中,信息素浓度的积累就会越慢;如果在 x 次迭代后,得到的最优路径比 L_2 要长, L_2 和 L_1 的差值就为负,说明当前迭代找到的最优路径不如之前寻找到的好,而根据式(12) L_2 和 L_1 的差值越大, $\Delta\tau_{ij}$ 就会越大,在之后的迭代中,信息素浓度就会加快挥发; L_2 和 L_1 的差值越小, $\Delta\tau_{ij}$ 就会越小,在之后的迭代中,信息素浓度挥发的就会越慢。这种动态调整是为了让算法能够及时修正迭代得到的结果,引导算法可以更快地找到最优的路径。

2.2.3 局部搜索策略

为了能够增加算法的开拓性,本文在所有车辆对路径选择完成之后选用 2-opt 搜索作为局部搜索的策略,而 2-opt 搜索只用于一辆车所选择的路径之中。

2.3 算法步骤

根据以上描述的改进策略,改进后的算法步骤如下:

1) 初始化参数。令迭代计数器初始值为 $N_c = 0$, 设置节点之间的距离为 $length[i][j](i, j = 0, 1, \dots, n)$, 迭代的最大次数为 N_c_CONST , 当前最优的路径长度 $shortest_len$, 客户服务的需求量 $load[j](j = 1, 2, \dots, n)$, 当前最优的路径表 $tabu_min$, 计算信息素的初值 τ_0 、转移吸引度 $\eta_{ij}(i, j = 1, 2, \dots, n)$ 以及负信息素 ϕ_{ij} 。将 e 辆车放置在车场,初始化每辆车的禁忌表 $ant[k].tabu(k = 1, 2, \dots, e)$, 所走的路径长度 $ant[k].length(k = 1, 2, \dots, e)$, 车辆的

负载量 $ant[k].load(k = 1, 2, \dots, e)$ 以及路径上的信息素 $\tau_{ij}(i, j = 1, 2, \dots, n)$ 。

2) 初始化车辆的位置。车辆按照式(8)选择客户,将所选择的客户添加到 $ant[k].tabu$ 中,并更新 $ant[k].length$ 和 $ant[k].load$ 的值。

3) 路径选择。在满足车辆负载条件的前提下,车辆按照式(9)进行客户的选择;否则车辆返回车场。将所选择的客户或车场添加到 $ant[k].tabu$ 中,更新 $ant[k].length$ 和 $ant[k].load$ 的值。

4) 信息素的局部更新。每当车辆作出一次选择,便根据式(10)对刚刚走过的路径 (i, j) 进行信息素的局部更新。

5) 2-opt 局部搜索。当所有车辆服务客户完成之后,寻找路径完毕时,对每辆车所寻找到的路径进行 2-opt 搜索,并更新 $ant[k].length$ 和 $ant[k].load$; 否则返回 3)。

6) 信息素的全局动态更新。统计当前迭代中的最短路径,将 $ant[k].length$ 与 $shortest_len$ 进行比较。如果 $ant[k].length < shortest_len$, 则用 $ant[k].length$ 替换 $shortest_len$, $ant[k].tabu$ 替换 $tabu_min$ 。根据式(11)和式(12)对当前最优路径进行信息素全局更新。

7) 迭代循环。如果 $N_c \leq N_c_CONST$, 则返回 Step 2, 开始新一次的迭代; 否则算法结束, 输出最优路径 $shortest_len$ 和最优路径表 $tabu_min$ 。

3 仿真实例

在车辆路由公共数据集^[13]中选择标准算例进行测试,使用 Matlab 进行编程,并在 CPU 为 Intel(R) Core(TM)i5-4460 CPU @ 3.20 GHz 3.20 GHz,内存 8

表 1 求解各案例结果

算例	客户数量	车辆数量	已知最优解	R	$P/\%$	平均值	运算时间/s
N19K2	18	2	212.65	212.65	0.00	214.72	3.11
N20K2	19	2	217.41	217.41	0.00	220.61	2.19
N21K2	20	2	212.71	217.48	2.24	221.33	5.02
N22K4	21	4	375.28	375.28	0.00	380.15	5.78
N40K5	39	5	461.72	468.25	1.41	471.36	13.24
N51K5	50	5	524.61	524.61	0.00	524.88	18.16
N16K8	15	8	451.35	451.35	0.00	456.36	1.56
N101K8	100	8	826.14	832.52	0.77	841.55	190.32
N23K8	22	8	531.17	531.17	0.00	532.70	3.65
N76K10	7	10	835.26	835.26	0.00	835.51	86.78
N151K12	150	12	1 028.42	1 038.22	0.95	1 049.87	833.11
N200K17	199	17	1 291.45	1 325.87	2.67	1 338.32	3 011.76

G 的计算机上运行。实验参数设置为： $\rho_1=0.1$ ， $\rho=0.1$ ， $\alpha=1$ ， $\beta=2$ ， $\gamma=1$ ， $\rho=15\sim30$ ， $r_0=0.8\sim0.95$ ， $NC=40$ 。

表 1 显示出运用本文算法求解选择出来的这部分算例的结果，求解每个算例都运行 10 次。这里设 B 为数据集中目前已知的算例最优解， R 为运用本文算法求解该案例得到的最优解， P 为运用本文算法求解案例得到的最优解对比数据集中已知最优解 B 的偏移率。从表 1 可以看出，运用给本文算法求解从数据集中随机抽选的 12 个算例中，有 7 个案例的求解结果和数据集中已知最优解相同，而在剩下 5 个结果有所偏差的案例中，案例 N200K17 的偏移率最大，但是偏移率仅为 2.67%。以上说明本文算法的性能是值得肯定的。

此外，从数据集中选择 5 个案例，运用本文算法与其他算法分别对案例进行求解，得出结果进行比较。表 2 显示了对比的结果。从表 2 可以看出，本文算法求解案例得到的最优解对比数据集中已知最优解的偏移率平均值为 0.88%，与其他 3 种算法相比，仅高于 HGA 0.03%。运用给本文算法求解从数据集中随机抽选的 5 个算例，有 2 个案例的求解结果和数据集中已知最优解相同。在求解运算的消耗时间方面，运用本文算法求解 N51K5、N76K10 所消耗的时间比 SA 要多一些，而求解另外 3 个案例的时间都是最短的，而且求解这 5 个案例所消耗的总时间最短，平均消耗时间也最短。

选择的 5 个案例，服务的车辆和被服务的客户

表 2 各算法对案例求解对比

算例	B		GA				HGA				SA				本文算法			
			R	$P/\%$	T/s		R	$P/\%$	T/s		R	$P/\%$	T/s		R	$P/\%$	T/s	
N51K5	524.61	524.81	0.04	213	524.61	0	23	524.61	0	6	524.61	0	18.16					
N76K10	835.26	849.77	1.74	765	838.89	0.43	617	844.31	1.08	78	835.26	0	86.78					
N101K8	826.14	840.72	1.76	1 148	829.47	0.4	717	832.32	0.75	228	832.52	0.77	190.32					
N151K12	1 028.42	1 055.85	2.76	2 475	1 034.8	0.62	1 961	1 061.55	3.22	1104	1 038.22	0.95	833.11					
N200K17	1 291.45	1 387.73	6.76	3 999	1 327.78	2.81	5 261	1 343.55	4.03	5 256	1 325.87	2.67	3 011.76					
平均值	-	-	2.61	1 720	-	0.85	1 715.8	-	1.82	1 334.4	-	0.88	828.03					

数量都比较多，问题的规模都很大，在这几个大规模问题中，相对于其他 3 种算法，本文算法无论是运算结果还是花费的时间都体现出了更佳的性能，而且问题的规模越大，本文算法体现出的性能越好。

4 结论

本文首先对 VRP 问题的理论进行了研究，引出 CVRP 问题，并且对其进行了数学模型的建立，然后在基本蚁群算法的基础上，改进算法使其能更好地对所建模型进行求解，最后从数据集中抽取案例，运用改进后的算法进行求解，并与其他算法进行比较，得出本文算法无论是运算结果还是花费的时间都体现出了更佳的性能和结论。

参考文献：

[1] 骆剑平,李霞,陈泯融.基于改进混合蛙跳算法的 CVRP 求解[J].电子与信息学报,2011,33(2):429-434.
[2] 张瑞锋,汪同三.新型遗传算法求解车辆路径问题研究[J].湖北大学学报(自然科学版),2012,34(2):239-242.
[3] DANTZING G, RASMER J. The truck dispatching problem [J].Management Science,1959(6):80-91.
[4] 陈印,徐红梅.混合算法在车辆路径优化问题中的应用

[J].计算机仿真,2012,29(25):357-358.
[5] 杨亚萍.基于车辆路径问题的建模及算法的研究[J].电脑开发与应用,2012,25(12):962-963.
[6] 杨中秋,张延华.改进蚁群算法在交通系统最短路径问题的研究[J].科学计算与信息处理,2003,32(8):76-78.
[7] 刘霞,杨超.最小-最大车辆路径问题的蚁群算法[J].解放军理工大学学报(自然科学版),2012,13(3):340-341.
[8] 张思亮,葛洪伟.求解车辆路径问题的一种混合方法[J].计算机工程与应用,2012,48(8):226-227.
[9] 李艳琴,张立毅,郭纯生,等.改进蚁群算法在应急 VRP 问题中的应用研究[J].数学的实践与认识,2012,42(9):133-140.
[10] KAJI T. Approach by ant tabu agents for traveling salesman problem [C]// In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,2001.
[11] KARABOGA N, KALINLI A, KARABOGA D. Designing digital IIR filters using ant colony optimisation algorithm[J]. Engineering Applications of Artificial Intelligence,2004,17(3):301-309.
[12] ROUX O, FONLUPT C, TALBI E G. Technical report LIL-98-04, laboratoired'informatique du littoral [D]. Calais, France:University du Littoral,1998.
[13] SOEDERKVIST O. The main algorithm research on financialtime series data mining [D].Linkoeeping:Linkoeeping University of Sweden,2001.