Optmization Letters manuscript No.
(will be inserted by the editor)

Optimizing Two-Level Reverse Distribution Networks with Hybrid Memetic Algorithms

Freitas, A. R. R. · Silva, V. M. R. · Campelo, F. · Guimarães, F. G.

the date of receipt and acceptance should be inserted later

Abstract In a Two-Level Reverse Distribution Network, products are returned from customers to manufacturers through collection and refurbishing sites. The costs of the reverse chain often overtake the costs of the forward chain by many times. With some known algorithms for the problem as reference, we propose a hybrid memetic algorithm that uses linear programming and a heuristic for defining routes. Moreover, we describe heuristics for deciding locations, algorithms to define routes for the products, and problem-specific genetic operators. Memetic algorithms have returned the best results for all instances.

Keywords Evolutionary Computation, Memetic Algorithms, Reverse Distribution Networks, Logistics

Freitas, A. R. R.
Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
E-mail: alandefreitas@gmail.com

Silva, V. M. R.
Instituto de Computação, Universidade Federal Fluminense
Niterói RJ 24210-240, Brazil
E-mail: victormrsilva@gmail.com

Campelo, F.
Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais
31270-010, Belo Horizonte, MG, Brazil
E-mail: fcampelo@ufmg.br

Guimarães, F. G.
Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais
31270-010, Belo Horizonte, MG, Brazil
E-mail: fredericoguimaraes@ufmg.br

1 Introduction

Reverse distribution networks involve the process of returning products to the manufacturer. It is different from the traditional forward chain as returned products often cannot be transported in the same manner as in the regular channel and the shipments usually have small quantities of products. It is an interesting problem as very often the cost of the reverse chain can overtake by many times the price of distributing the product to customers [1,2].

A good reason for studying this problem is its environmental impacts. The products being returned often include products that do not work anymore (such as electronic components), need to be recycled (such as mercury-added lamps) or are even dangerous (such as batteries). Our work is based on a formulation of two-level reverse distribution networks [3], in which the first level has collection sites, where the products can be temporarily stored until they are sent to refurbishing sites, represented by the second level (Section 2).

Many heuristics have been proposed for this NP-Complete problem [3,4]. In this paper we propose memetic and genetic algorithms to compare with those other approaches. Genetic and Memetic algorithms are used to analyze the facilities while an external method, based either on linear programming or heuristics, is employed to calculate good routes, forming a hybrid algorithm.

The problem model considers the cost and advantages of opening possible facilities as well as the cost of transporting the products (Section 3).

Given a formulation for the problem, we present in this work:

- Heuristics for deciding open facilities and routes (Section 4).
- Problem-specific evolutionary algorithms (Section 5).
- A set of 25 instances and an analysis of the search space (Section 6).
- Experiments with all algorithms (Section 7).

Conclusions and future work are reported in Section 8.

2 Two-Level Reverse Distribution Networks

Reverse Logistics is the process of returning products to the manufacturer because they need to be repaired or recycled. This reverse chain is very different from the traditional forward chain as returned products often cannot be transported in the same manner as in the regular channel and the shipments usually have small quantities of products. Therefore, reverse costs may be higher than moving the product to the consumer [5]. Usually with specific models for the problem, some works consider both the forward and reverse chain [6,7,5,8–11] as others consider only the reverse chain [12,13,3].

There are many models [5,14,12] and solutions [5,13,6,10,11,7,9,15] for this problem. The formulation used in this work [3] can be reduced to a NP-complete problem [4], and therefore, the difficulty of finding an optimal solution for large instances in reasonable time justifies the use of heuristics for the problem. The best heuristics proposed for this formulation of reverse logistics so far are concentrations heuristics [3] and evolutionary algorithms [16].

3 Representing the Problem and its Solutions

The general model for the problem [3] is described as follows:

- $I\{i/i\} \rightarrow$ origination, $J\{j/j\} \rightarrow$ collection, $K\{k/k\} \rightarrow$ refurbishing
- $C_{ijk} \rightarrow$ cost of transporting an unit from i to k through j
- $F_j \rightarrow$ cost of opening the collection site j
- $G_k \rightarrow$ cost of opening the refurbishing site k
- $a_i \rightarrow$ number of products at the origination site i
- $B_j \rightarrow$ capacity of the collection site j
- $D_k \rightarrow$ capacity of the refurbishing site j
- $P_{min}, P_{max} \rightarrow$ minimum and maximum number of open collection sites
- $Q_{min}, Q_{max} \rightarrow$ minimum and maximum number of open refurbishing sites

Each solution is represented by:

- $X_{ijk} \rightarrow$ fraction of units at originating site i transported through j and k ($j = 0$ is used to indicate that the products go straight to k)
- $P_j \rightarrow$ boolean representing whether the collection site j is open
- $Q_k \rightarrow$ boolean representing whether the refurbishing site j is open

Our objective is to minimize $\sum_i \sum_j \sum_k C_{ijk} a_i X_{ijk} + \sum_j F_j P_j + \sum_k G_k Q_k$ subject to:

$$\sum_j \sum_k X_{ijk} = 1 \text{ for all } i; \qquad 0 \leq X_{ijk} \leq 1 \text{ for all } i, j, k$$

$$\sum_i \sum_k a_i X_{ijk} \leq B_j \text{ for all } j; \qquad \sum_i \sum_j a_i X_{ijk} \leq D_k \text{ for all } k$$

$$X_{ijk} \leq P_j \text{ for all } i, j, k; \qquad X_{ijk} \leq Q_k \text{ for all } i, j, k$$

$$P_{min} \leq \sum_j P_j \leq P_{max} \text{ for all } j; \qquad Q_{min} \leq \sum_k Q_k \leq Q_{max} \text{ for all } j$$

$$P_j \in \{0, 1\} \text{ for all } j; \qquad Q_k \in \{0, 1\} \text{ for all } k$$

4 Heuristics

Initially, we have the first subproblem of selecting which facilities will be open (P_j and Q_k). This must be done in a way to minimize, besides the cost of sending products, the cost of opening the required sites. Then, we must find the best routes in the network for all the products to solve the second subproblem. The fraction of products sent through each possible route is represented by the variable X_{ijk}, where i is the origin, j is an open collection site, and k an open refurbishing site. There may be cases in which it is better to send a product directly to a refurbishing site k. This is represented by X_{i0k}.

For the first subproblem, we consider two heuristics. The first one is based on a greedy principle and the second one is inspired by an original solution for the p-median problem [17,3]:

Greedy Heuristic (i) Rank order all collection sites P_j and refurbishing sites Q_k by their capacity/cost. (ii) Sort an integer between p_{min} and p_{max} and save it as p_{goal}. Sort an integer between q_{min} and q_{max} and save it as q_{goal}. (iii) The p_{goal} collection sites with highest capacity/cost value are open. The q_{goal} collection sites with highest capacity/cost value are open.

Concentrations Heuristic (i) For t iterations (in this work, experimentally set to $t = 100$), a subset of size P_{max} of collection sites and Q_{max} refurbishing sites is randomly selected and the routing of products for these sites is solved by another algorithm. All solutions are saved and the best solution is marked. (ii) The sites used the most in the best solutions are added to the best solution found in the random selection phase in order to form a new solution, and the routing subproblem is solved to optimality. This new solution is compared to the best solution from the random selection phase. (iii) Add each unused site to the best solution and solve the problem. If a better solution is found, remember this solution and its configuration, but leave the best solution unchanged. Check if any of the other unused sites give a better solution than the new solution found. Repeat this until all the unused sites are checked. Stop when no better solution is found.

The heuristics can generate solutions with different numbers of open facilities in the two levels of the model. These heuristics define which facilities will be opened. For the second subproblem, the routing problem, we present a heuristic with a greedy principle and a numeric solution:

Routing Heuristic For each origin site to be examined: (i) Among the valid routes for the products, find the one with lowest cost. (ii) Send as many products as possible through this route and update its capacity. (iii) If there are still products in the origination site, send them through the next best route. Otherwise, examine the next origin site.

Linear Programming Routing (i) Given that the open facilities are already defined, eliminate the parts dependent on P and Q, resulting on a linear programming problem. (ii) Solve the new subproblem with a linear programming (LP) algorithm.

The routing heuristic does not guarantee the optimal but it has the advantage of being considerably less computationally expensive than exact methods. The formulation of the subproblem for the LP routing is defined as minimizing $\sum_i \sum_j \sum_k C_{ijk} a_i X_{ijk}$ subject to $\sum_j \sum_k X_{ijk} = 1$ for all $i; \sum_i \sum_k a_i X_{ijk} \leq B_j$ for all $j > 0; \sum_i \sum_j a_i X_{ijk} \leq D_k$ for all $k; 0 \leq X_{ijk} \leq 1$ for all i, j, k.

5 Hybrid Memetic Algorithm

5.1 Genetic Algorithm

In our genetic algorithm, each individual is represented by two binary vectors that represent P and Q. The constraints related to P_{min}, P_{max}, Q_{min}, and

Q_{max} are all satisfied by the genetic operators, which do not allow bit-flops exceeding these limits. Once the values P and Q are found, the routing heuristic or the LP routing finds the route for that solution before calculating its objective function value. If the capacities of the facilities are not enough to keep the products from the origin sites, the following objective function is used:

$$\text{minimize } f(x) = \left(\sum_i 1 \sum_j 1 \sum_k 1\right) \max(C_{ijk}) \max(a_i) + c$$

$$\text{where } c \quad = \max\left(0, \sum_j B_j P_j\right) + \max\left(0, \sum_k D_j Q_j\right)$$

This objective function guarantees that unfeasible solutions always have worse fitness values than feasible solutions. Besides the fitness functions, genetic operators are important mechanisms to implicitly guide the search and intelligently filtrate solutions [18]. Having that in mind, the crossover operator keeps the open facilities that would probably lead to a good fitness. That is done with the crossover operator exemplified in Figure 1 and formally defined as:

1. The number of open facilities *goal* of the child is an integer proportional to the fitness fit of the parents P in the form $\lfloor(\frac{fit_1|P_1|+fit_2|P_2|}{fit_1+fit_2}) + 0.5\rfloor$.
2. $A = P_1 \cup P_2$ and $B = P1 \cup P_2 - P_1 \cap P_2$.
3. Add randomly chosen elements from B to A until $|A| = goal$.

The mutation operator is a bit-flop in P or Q that only generates new solutions that respect the limits P_{min}, P_{max}, Q_{min}, and Q_{max}. The crossover and mutation probabilities are adapted [19] every generation for each individual. In each generation, the whole population is replaced by their children. The routes X for new individuals are calculated with a LP algorithm (Section 4), as it was better than the routing heuristic in preliminary experiments [16].

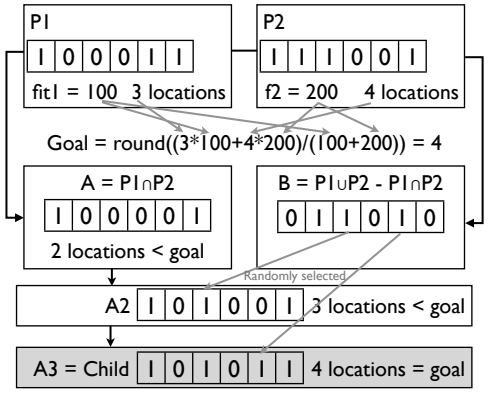


Fig. 1 Crossover Operator

The final fitness of the individuals is given by their rank [20] instead of their objective function values and stochastic ranking selection [21] is used to decide which parents will generate children.

5.2 Memetic Algorithm

Memetic algorithm is a term usually employed for population methods with local improvement procedures. In our memetic algorithm for the proposed problem, after calculating the route X for the products, we apply a local search to each individual to look for better solutions in the neighborhood.

A neighbor solution is a solution generated by changing one value of P or Q with respect to the constraints involving P_{min}, P_{max}, Q_{min} and Q_{max}.

A first improvement local search looks for better neighbor solutions. All neighbors are examined in random order, until a better neighbor is found.

5.3 Hybridization

By putting all the concepts together, we developed a hybrid memetic algorithm based on the genetic algorithm and heuristics described before. As for the genetic algorithm, it can use initial random solutions, greedy solutions or solutions from the heuristics concentration. The routes for the solutions can be calculated with the LP routing or the routing heuristic. However, there is evidence that it is not worth using only the heuristics to estimate the best route, despite the computational cost of the LP algorithm [16]. In this previous work, genetic algorithms with the routes based on heuristics converged prematurely to median solutions in relation to LP routing solutions.

The hybrid method that we propose involves all those algorithms. The genetic algorithm is used to decide the locations, the routing heuristic is used to calculate a good route. The local search uses the routing heuristic to check all the neighbors.

As the LP algorithm is always going to return better results than the routing heuristic, the refinement decreases the maximum objective value a solution can have after the LP algorithm. When the local search can no longer find better neighbors, a LP algorithm is used to calculate the best route possible. The steps of the local search are:

1. While there is improvement in the current solution.
   (a) Randomly choose a valid neighbor on P or Q, that was not tested yet.
   (b) Calculate X for the neighbor with the routing heuristic.
   (c) If the neighbor is better than the current solution.
      i. The neighbor becomes the current solution.
      ii. Go to the beginning to test the new neighbors.
2. Calculate X for the final solution with the LP routing algorithm.

Table 1 Instances

(a) Location Values

F_j	$0.1([1, 10000] + B_j[0, 10])$
G_k	$0.1([1, 25000] + B_j[0, 100])$
C_{ijk}	Distance from i to j to k
a_i	$[0, 500]$
B_j	$[0, 6000]$
D_k	$[0, 30000]$

(b) Instance Sets

Set	$\|I\|$	$\|J\|$	P_{max}	$\|K\|$	Q_{max}	Complexity
1	30	14	4	12	2	114738
2	40	20	6	15	4	117292400
3	50	30	6	20	4	4.7591×10^{9}
4	70	30	6	20	4	4.7591×10^{9}
5	100	40	8	30	6	7.6934×10^{13}

6 Instances

For comparison, we used 25 instances from a previous work [16]. The instances are all available for download[1] along with their best objective function values obtained with CPLEX. The instance values are based on locations distributed on a 100×100 square. The 25 instances are divided in 5 sets, described in Table 1 (the notation $[n_1, n_2]$ represents a uniformly sampled random number between n_1 and n_2). The complexity of each instance, measured as the number of feasible combinations of P and Q values, is also represented in the table.

7 Experiments

We compared four non-evolutionary heuristics, three genetic algorithms and three memetic algorithms, as described in Table 2. The abbreviations used were Greedy Heuristic (`Gr`), Concentrations Heuristic (`CH`), Routing Heuristic (`RH`), LP Routing (`SR`), Genetic Algorithms (`GA`) with LP Routing, Memetic Algorithms (`MA`) with local search based on Routing Heuristic and LP Routing, Random Solution Generation (`RS`). Methods A-D are the non-evolutionary heuristics that generate solutions and routes. Methods F, H, J are Genetic Algorithms with different heuristics for generating the initial solutions (`RS`, `Gr` or `CH`) and using LP Routing to calculate the best routes. LP routing was shown to be significantly more efficient than heuristic routing when embedded in a genetic algorithm in a previous work [16]. Methods E, G, I are Memetic Algorithms with different heuristics for generating the initial solutions (`RS`, `Gr` or `CH`). Ten replicates of the experiment were performed on each instance with a time limit of 30 seconds for methods E-J. The constructive methods A-D have their own stopping criteria according to the algorithms described in

[1] All results, solutions, instances, and figures are available online at `http://www.alandefreitas.com/downloads/problem-instances.php`

Table 2 Methods Compared

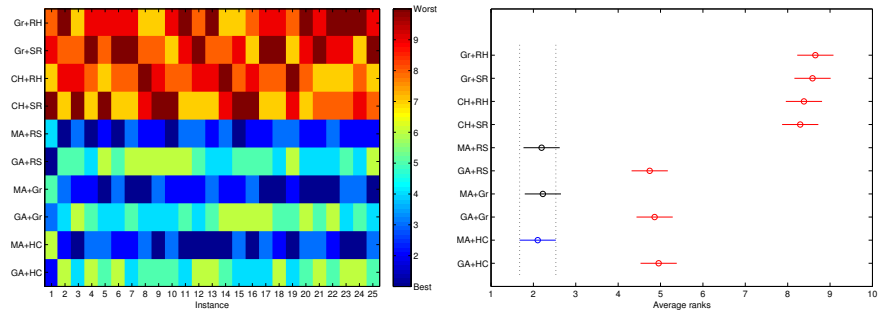
Method	A	B	C	D	E
Heuristics	`Gr+RH`	`Gr+SR`	`CH+RH`	`CH+SR`	`MA+RS`
Method	F	G	H	I	J
Heuristics	`GA+RS`	`MA+Gr`	`GA+Gr`	`MA+CH`	`GA+CH`

Section 4. For all tests performed, the predefined significance threshold was set as 95%.

Figure 2(a) shows the difference of mean objective value obtained by the methods in each instance. For better visualization, the values were scaled by average rank. The results of the Friedman test were highly significative ($p < 10^{-4}$), indicating that at least some of the methods tested presented differences on their performance values across the instances tested.

Tukey's HSD method [22] was employed for post-hoc analysis. The results of the pairwise comparisons of the ranks for each method are presented in Figure 2(b). The methods tested are divided into three main groups: heuristics (methods A-D), genetic algorithms (methods F, H, J), and memetic algorithms (methods E, G, I). The mean rank values of the methods (from A to J) were 8.654, 8.586, 8.384, 8.296, 2.194, 4.744, 2.224, 4.86, 2.102, 4.956. Although method I (memetic algorithm with concentrations heuristic) was the one with best rank value (2.102), we cannot affirm significant difference between any of the memetic algorithms at confidence level $\alpha = 0.05$. The memetic algorithms, however, were all significantly better than their purely genetic equivalents, which in turn were all significantly superior to the heuristics tested.

In order to have a parametric reference of the quality of the solutions obtained, all the instances were solved to optimality with CPLEX. Figure 3 is a reference of the deviation in relation to the best solutions for each instance with all the algorithms. Figure 3(a) shows an absolute comparison between all methods where while Figure 3(b) shows the results for the evolutionary methods only. All those figures and results are also available online.



(a) Rank of the average results for each method

(b) Estimated average ranks and confidence intervals for the methods tested

Fig. 2 Comparison between the methods

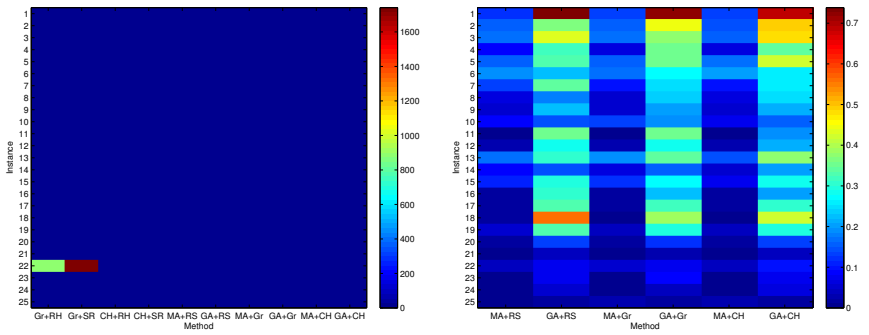
8 Conclusion and Future Work

Given the results of the experiment, we conclude that memetic algorithms are significantly better than other approaches for two-level reverse distribution problem in the conditions applied here. However, differences in the generation of the initial population did not influence the performance of any evolutionary algorithm (genetic or memetic) at confidence level $\alpha = 0.05$. Given these results, it is reasonable to assume that the most parsimonious way to initialize the evolutionary approaches presented here would be to use a random initial population, since it would imply the least computational effort and achieve results that could not be proven to have statistical difference.

Interesting avenues for further exploration include the investigation of the influence of different parameters for the memetic algorithms; the determination of globally optimal solutions for the proposed instances, in order to allow for different performance metrics - e.g., *time to target* - to be implemented in future tests; implementation and comparisons with algorithms using different genetic operators; implementation of parallel evaluation of candidate solutions, in order to speed up the tests and the application of the proposed algorithms; tests on larger instances and on application-driven problems; and investigation of the effects of the stop criterion (time limit) in the performance of the algorithms.

9 Acknowledgments

This work has been supported by the Brazilian agencies CAPES, CNPq (grants 472446/2010-0 and 305506/2010-2), and FAPEMIG (grant APQ-04611-10); and by the Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.



(a) All Methods (b) Evolutionary Methods

Fig. 3 Deviation from the best solution

References

1. Min, H.: A bicriterion reverse distribution model for product recall. Omega **17**(5) (1989) 483 – 490
2. Chandran R., L.R.: Product recall: A challenge for the 1980s. International Journal of Physical Distribution & Logistics Management **11**(8) (46–55 1981) 483–490
3. Jayaraman, V., Patterson, R.A., Rolland, E.: The design of reverse distribution networks: Models and solution procedures. European Journal of Operational Research **150**(1) (2003) 128 – 149
4. Davis, P., Ray, T.: A branch-bound algorithm for the capacitated facilities location problem. Naval Research Logistics Quarterly **16**(3) (1969) 331–343
5. Fleischmann, M., Bloemhof-Ruwaard, J.M., Dekker, R., van der Laan, E., van Nunen, J.A., Wassenhove, L.N.V.: Quantitative models for reverse logistics: A review. European Journal of Operational Research **103**(1) (1997) 1 – 17
6. Hu, T.L., Sheu, J.B., Huang, K.H.: A reverse logistics cost minimization model for the treatment of hazardous wastes. Transportation Research Part E: Logistics and Transportation Review **38**(6) (2002) 457 – 473
7. Sheu, J.B., Chou, Y.H., Hu, C.C.: An integrated logistics operational model for green-supply chain management. Transportation Research Part E: Logistics and Transportation Review **41**(4) (2005) 287 – 313
8. Montané, F.A.T., Galvão, R.D.: A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Computers & Operations Research **33**(3) (2006) 595 – 619
9. Ko, H.J., Evans, G.W.: A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3pls. Computers & Operations Research **34**(2) (2007) 346 – 366
10. Alshamrani, A., Mathur, K., Ballou, R.H.: Reverse logistics: simultaneous design of delivery routes and returns strategies. Computers & Operations Research **34**(2) (2007) 595 – 619
11. Lu, Z., Bostel, N.: A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. Computers & Operations Research **34**(2) (2007) 299 – 323
12. Spengler, T., Püchert, H., Penkuhn, T., Rentz, O.: Environmental integrated production and recycling management. European Journal of Operational Research **97**(2) (1997) 308 – 326
13. Bautista, J., Pereira, J.: Modeling the problem of locating collection areas for urban waste management. an application to the metropolitan area of barcelona. Omega **34**(6) (2006) 617 – 629
14. Barros, A., Dekker, R., Scholten, V.: A two-level network for recycling sand: A case study. European Journal of Operational Research **110**(2) (1998) 199 – 214
15. Horvath, P.A., Autry, C.W., Wilcox, W.E.: Liquidity implications of reverse logistics for retailers: A markov chain approach. Journal of Retailing **81**(3) (2005) 191 – 203
16. Freitas, A., Silva, V., Guimarães, F., Campelo, F.: Genetic algorithms applied to reverse distribution networks. In Snášel, V., Abraham, A., Corchado, E.S., eds.: Soft Computing Models in Industrial and Environmental Applications. Volume 188 of Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg (2013) 317–326
17. Rosing, K., ReVelle, C.: Heuristic concentration: Two stage solution construction. European Journal of Operational Research **97**(1) (1997) 75 – 86
18. Freitas, A.R., Guimarães, F.G.: Originality and diversity in the artificial evolution of melodies. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. GECCO '11, New York, NY, USA, ACM (2011) 419–426
19. Whitacre, J.M.: Adaptation and self-organization in evolutionary algorithms. CoRR **abs/0907.0516** (2009)
20. Kreinovich, V., Quintana, C., Fuentes, O.: Genetic algorithms: What fitness scaling is optimal? Cybernetics and Systems **24**(1) (1993) 9–26
21. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation **4** (2000) 284–294
22. Hsu, J.: Multiple Comparisons. Chapman and Hall (1996)