A Systematic Heuristic Rules Analysis Methodology for Routing Problems

D. Crespo-Pereira[1], D.del Río-Vilas[2], J. L. Crespo-Mariño[3], A. García-del-Valle[4]
[1, 2, 3, 4]Grupo Integrado de Ingeniería, University of A Coruña, Ferrol, Spain.
(dcrespo, davidelrio, jcm, agvalle@udc.es)

Abstract - **Heuristic constructive algorithms have been widely and successfully applied to the solution of routing problems. Since they generally consist of an iterative insertion of nodes to construction routes, prioritization rules for assignments is critic for algorithm's performance. Developing these rules is time consuming and relies much on researcher skills and knowledge on problem features. This paper proposes a systematic methodology for a widespread exploration of prioritization rules aiming at reducing human effort on its development. A general model for prioritization is achieved by means of an artificial neural network. Parameters are tuned for the specific problem by an evolutionary strategy search. The methodology is formulated for generic routing problems and applied to VRPTW to illustrate its operation and as a preliminary assessment of its capabilities. Neural networks are evolved for Solomon's benchmark instances and analyzed to gain knowledge on underlying rules.**

Keywords - **Prioritization rules, node assignments, VRPTW, neural network, evolutionary strategy.**

I. INTRODUCTION

Routing problems constitute one of the most well known and characteristic problems in operations research. They play a central role in logistics due to the relatively high importance of distribution costs in supply chain [1].

In general terms, routing problems deal with finding an optimal or quasi-optimal set of routes for rendering the corresponding services - freight transportation, repair services, passengers transportation, etc. - to a given set of customers or suppliers geographically distributed and connected by a transportation network. But most of routing problems are NP-complete combinatorial, which implies high computational costs when aiming to find optimal solutions. This has lead to a widespread development of heuristic and metaheuristic approaches, more suitable for real applications.

Some of the most classical and studied problems include the TSP, the CVRP or the one in which this work is focused, the VRPTW [2]. Common approaches to solve routing problems include optimal algorithms, heuristic algorithms and metaheuristics.

Currently, main trends in routes optimization research include the definition of new problems closer to applications in real environments [3], research of new hybrid algorithms with higher performance and generalization capabilities [4] and the improvement of existing solution techniques.

In this paper, a hyper-heuristic methodology [5] is introduced in order to develop a systematic procedure for the exploration of heuristic constructive rules. Our methodology is aimed to be suitable for general routing problems in real environments. It is applied to the solution of the VRPTW as an initial study to evaluate its feasibility and to identify the main aspects for its future improvement.

II. METHODOLOGY

In this section, a generic routing problem is formulated according to notation used in [6]. A generic routing problem is an optimization problem that consists of finding a set of routes that minimizes a cost function given a transportation network and the problem constraints. Let $G=\{V,E\}$ be the graph of the transportation network where V denotes the set of nodes. A route is a walk in V: $R=\{v_1,a_1,\dots v_{n-1},a_{n-1},v_n; v_i\in V; a_i=(v_i,v_{i+1})\}$. It might be closed or not depending on problem constraints. $|R|$ is the number of nodes in a route.

Let R_G be the set of all routes in G. The cost function of a route is a function $CF: R_G\to\mathbb{R}$. A solution S to the routing problem is a set of routes that verifies problem constraints and Ψ the set of all solutions. A solution cost is the sum of its routes costs $c(S)=\sum_\alpha CF(R_\alpha); R_\alpha\epsilon S; \alpha\in 1,\dots,|S|$. A route R is feasible if $\exists S\in\Psi/R\in S$. A solution is optimal if its cost is lower than or equal to every other solution: $c(S^*)\le c(S)\ \forall\ S\in\Psi$. A routing problem aims at finding S^*.

An assignation is defined as a pair node-route. $A_{\alpha i}=A(R_\alpha,v_i)=\{R_\alpha,v_i/v_i\notin R_\alpha\}$. Every assignation defines an extended route of R_α as the route formed by adding the node at the end of the route. $R'_\alpha=\{v_{\alpha 1},\dots,v_{\alpha n},v_i; v_{\alpha j}\in R_\alpha\}$. The cost of an assignation is a function that represents the difference between the cost of the initial and extended routes. $CF(A_{\alpha i})=CF(R'_\alpha)-CF(R_\alpha)$. Every route can be obtained as a sequence of assignations starting from an empty route.

$$R_\alpha=\{v_{\alpha 1},\dots,v_{\alpha n}\}=\{A(\emptyset,v_{\alpha 1}),\dots,A(\{v_{\alpha 1},\dots,v_{\alpha n-1}\},v_{\alpha n})\} \quad (1)$$

An assignation is said feasible if the extended route is feasible. A feasibility function for every assignation is such that its value is 1 if it is feasible and 0 if not $fs_{\alpha i}=\begin{cases}1\leftrightarrow R'_\alpha\epsilon FR\\ 0\ otherwise\end{cases}$.It can be easily proved that a route is feasible if and only if all the assignations necessary to obtain it are so.

Since every route can be decomposed into a sequence of assignations, a solution can be also regarded as a set of feasible assignations. A partial solution is a set of assignations such that any solution contains all of them. The feasibility of an assignation given a partial solution depends on the latter one.

Constructive algorithms [7] are based on a sequential performance of assignations. Initially, all the vertices remain unassigned and the construction routes are empty. In every step, a set of feasible assignations is performed and thus the construction routes extended with non-assigned vertices. Once an assignation is performed, the set of solutions that can be achieved by the algorithm is reduced and consequently the solution found might not be optimal. Thus, the prioritization of the feasible assignations in each step is critical.

The prioritization rules can be modeled by a prioritization function of a set of factors that characterize the assignation: $p_{\alpha i} = PR(f_1(A_{\alpha i}), \dots, f_{NF}(A_{\alpha i}))$, where $f_j(A_{\alpha i})$ is each one of the assignation factors and *NF* the total number. These factors should be specifically defined for each problem. Ties in priority might be broken randomly or by a proper selection of the model.

The pseudo-code for a constructive algorithm is the following:

L_0 is the set of non-assigned vertices.

CR is the set of construction routes.

1. $CR = \emptyset$.
2. Search assignation $A_{\alpha^* i^*}$ such that $(fs_{\alpha i} = 1) \cap (p_{\alpha^* i^*} \geq p_{\alpha i} \; \forall \; \alpha \in 1, \dots, |CR|; i \in 1, \dots, |L_0|)$.
3. Add vertex v_{i^*} to route R_{α^*}.
4. Remove v_{i^*} from L_0.
5. If $L_0 \neq \emptyset$ return to 2.

The basis of our method is the substitution of predefined prioritization rules for a parameterized model $p_{\alpha i} = PM(f_1(A_{\alpha i}), \dots, f_{NF}(A_{\alpha i}), \gamma_1, \dots, \gamma_m)$, where $\gamma_1, \dots, \gamma_m$ represents the parameters of the model. Due to parameterization, optimization techniques can be applied for model tuning.

The methodology involves the following steps given a specific problem:

1. Define constraints and feasible assignations.
2. Select a set of assignation factors which are determinant for algorithm's performance.
3. Choose a parameterized model for the priority evaluation.
4. Adjust model parameters by an optimization technique.
5. Check possible correlations among factors in order to simplify the model.
6. Check model's sensitivity to the factors and remove those which lack a significant effect.
7. Analyze obtained rules.

Appropriate selection of the input factors to the model is necessary for a good performance. This step relies much on the researcher experience. However, it might always be chosen an initial wide set of factors to be further improved by analyzing the model. Once the factors are selected, the model should be capable to represent a wide set of rules in order to perform an intensive exploration.

Our model for the prioritization function is a neural network. Specifically, we use a multilayer perceptron with sigmoid activation function due to its capability to approximate every real continuous function, and also non continuous if used more than one hidden layers [8]. The parameters of the model are the weights and bias of the connections between neurons. The input factors are normalized into the range 0-1. The output of the net is the priority.

Given a set of model parameters, the solution to the routing problem is unique, so that it is possible to define a cost function of the parameters as $c(\gamma_1, \dots, \gamma_m) = \sum_\alpha CF(R_\alpha)\, ; R_\alpha \epsilon\, S$. Thus routing problem has been transformed into searching for the parameters vector that minimizes the cost. Since two models' cost can only be different because of producing different assignations, the cost function is a non-continuous function. Cost remains constant in regions which lead to equal solutions.

Evolutionary search is an optimization technique appropriate for optimizing the neural network model. Gradient methods are unavailable since objective function is stepped. The number of parameters in a neural network is high so the optimization problem is highly dimensional. Evolutionary search allows a wide exploration of parameters and avoid local optimal solutions.

Once a model has been evolved, it comes the time to analyze its properties to improve it and gain information about the produced rules. Input factors may present correlations that make the model more complex without providing extra information. Thus, independence tests might be applied among the components of the inputs generated in the problem solution. A test for independence based on an empirical Copula process [9] has been used. This is a non-parametric test suitable for multidimensional independence analysis. It allows not only analyzing the independence between pairs of variables, but also the joint correlation among all the subsets of variables.

The model's sensitivity to input variables is analyzed by means of the priority function partial derivative with respect to each input at every vector generated in the solution. The distribution of this derivative on generated data gives us the factor's relevance.

A complementary way to check the effect of a factor on the priority is to plot the joint distribution of each factor and the priority for data generated in the solution. This can also aid to check the normalization of factors.

Finally, it is proposed to obtain a set of plots of the priority as a function of each factor for random values of the other factors. This is helpful when studying the different behaviors that the model may produce.

III. APPLICATION TO VRPTW

The VRPTW is a generalization of the capacitated vehicle problem in which time constraints are defined for visiting the nodes. Let n_i be the node identified by subscript i and N the number of nodes. Then q_i is the demand of node n_i, to_i the opening time at n_i, tc_i the closing time at n_i, ts_i the service time, d_{ij} the distance between n_i and n_j, and t_{ij} the time to travel from n_i to n_j. No assumptions about distances or travel times are imposed so that the following procedure might be applied to problems with non-Euclidean distances or asymmetric distance matrix.

The constraints for solution routes involve the following, where:

R_k is the route for which node n_i is assigned.

The arrival time in the route to n_i is ta_i.

The departure time in the route from n_i is td_i.

$$\sum_{j=1}^{j=|R_k|} d_{n_j} \leq C \quad (2)$$

$$ta_i \leq tc_i \quad (3)$$

$$td_i \geq ta_i + ts_i \quad (4)$$

The set of factors used as an input for the prioritization model was intended to be wide to provide new and complex rules. A set of sixteen parameters were selected. To introduce these parameters, it is used the following notation:

Given a sorted list L and its element e_i, then the order of the element is its position in the list sorted from minimum to maximum and it is noted as $O(e_i, L)$.

The assignation with subscript α for which the parameters are defined is the pair route-node (R_k, v_i).

The set of feasible assignations in the given step is LA.

The route head node (RH) is the last node of a given route.

The direct gain of an assignation is the gain obtained by performing an assignation compared to the second best possible assignation to the route. Thus:

$$dg_{k,i} = \min_j\{d_{RH_k,j}; d_{RH_k,j} \neq \min_l\{d_{RH_k,l}\}\} - d_{RH_k,i} \quad (5)$$

The total gain of an assignation is the direct gain obtained by performing an assignation compared to the second best direct gain to assign the node.

$$g_{k,i} = dg_{k,i} - \max_j\left\{dg_{RH_j,i}; dg_{RH_j,i} \neq \max_l\{dg_{RH_l,i}\}\right\} \quad (6)$$

The normalization waiting factor (WNF) is used to normalize waiting times and can be set up according to characteristic time units in the problem.

Direct degree of a node $\delta(v_i)$ is the number of feasible assignations to a route ended by it.

Inverse degree of a node $\delta'(v_i)$ is the number of feasible assignations to a route ended by it.

1. Relative total gain for all the selected assignations.
$$RTG_{k,i} = \frac{g_{k,i}}{\max_{A\in LA} g(A)} \quad (7)$$
2. Relative total gain to route.
$$RGR_{k,i} = \frac{g_{k,i}}{\max_j g_{k,j}} \quad (8)$$
3. Waiting factor at the assigned node.
$$wt_{k,i} = \begin{cases} \frac{1}{1+\frac{to_i - ta_i}{WNF}} \\ 1 \; otherwise \end{cases} \quad (9)$$
4. Arrival time factor.
$$TAF_{k,i} = \frac{ta_i}{\max_j tc_j} \quad (10)$$
5. Centrality factor.
$$CNF_i = \frac{d_{0,i}}{\max_j d_{0,j}} \quad (11)$$
6. Centrality order.
$$CNO_i = O\left(d_{0,i}, \{d_{0,j}; j \in 1, \dots, N\}\right) \quad (12)$$
7. Closing time factor.
$$CTF_i = \frac{tc_i}{\max_j tc_j} \quad (13)$$
8. Closing time order.
$$CTO_i = O\left(ct_i, \{ct_j; j \in 1, \dots, N\}\right) \quad (14)$$
9. Inverse degree factor.
$$IDF_i = \frac{\delta'(i)}{N} \quad (15)$$
10. Direct degree factor.
$$DDF_i = \frac{\delta(i)}{N} \quad (16)$$
11. Assignation demand factor.
$$ADF_{k,i} = \frac{\sum_{j\in R_k} q_j + q_i}{C} \quad (17)$$
12. Assigned rate.
$$AR = \frac{|L_0|}{N} \quad (18)$$
13. Centrality factor of route.
$$CNFR_{CR_k} = \frac{d_{0,CR_k}}{\max_j d_{0,CR_k}} \quad (19)$$
14. Centrality order of route.
$$CNOR_{CR_k} = O\left(d_{0,CR_k}, \{d_{0,CR_k}; j \in 1, \dots, N\}\right) \quad (20)$$
15. Route load factor.
$$RLF_k = \frac{\sum_{j\in CR_k} q_j}{C} \quad (21)$$
16. Direct degree fraction of the route.
$$DDF_k = \frac{\delta(CR_k)}{N} \quad (22)$$

IV. EXPERIMENTATION AND RESULTS

The algorithm has been coded in JAVA. As benchmark of problems to evaluate this methodology, we employ the Solomon Instances. In order to reduce the time necessary for models evolution, only the first 25 nodes plus the depot of each instance are considered.

As benchmark of problems to evaluate this methodology, we employ the Solomon Instances. In order to reduce the time necessary for models evolution, only the first 25 nodes plus the depot of each instance are considered.

The multilayer perceptron used has one hidden layer with 4 neurons. Slopes of activation functions are fixed to 0.5 with the aim to reduce the number of parameters in the model. This means that a set of 73 parameters are needed to code the net weights and bias. Maximum and minimum weight values and bias are limited to +8 and -8 respectively.

TABLE I
RESULTS TO SOLOMON 25 NODES INSTANCES

Case	Distance	Dif. to optimal*	Case	Distance	Dif. to optimal*
RC101	474.40	2.82%	R201	467.70	0.95%
RC102	362.66	3.09%	R202	435.84	6.17%
RC103	339.00	1.86%	R203	424.76	8.52%
RC104	317.38	3.52%	R204	373.15	5.11%
RC105	423.11	2.87%	R205	412.40	4.94%
RC106	356.73	3.25%	R206	394.91	5.48%
RC107	299.50	0.40%	R207	393.83	8.91%
RC108	296.83	0.79%	R208	357.97	9.07%
RC201	364.01	1.06%	R209	386.23	4.19%
RC202	359.37	6.32%	R210	415.31	2.65%
RC203	356.40	9.02%	R211	355.64	1.35%
RC204	324.32	8.22%	C101	191.81	0.27%
RC205	355.59	5.20%	C102	190.74	0.23%
RC206	333.33	2.88%	C103	201.54	7.83%
RC207	309.63	3.80%	C104	194.58	1.72%
RC208	279.61	3.91%	C105	191.81	0.27%
R101	618.33	0.20%	C106	191.81	0.27%
R102	578.44	5.73%	C107	191.81	0.27%
R103	479.86	5.56%	C108	191.81	0.27%
R104	461.74	10.7%	C109	191.81	0.27%
R105	531.80	0.25%	C201	215.54	0.39%
R106	498.96	7.21%	C202	215.54	0.39%
R107	446.67	5.27%	C203	215.54	0.39%
R108	430.96	8.47%	C204	222.27	4.30%
R109	442.63	0.30%	C205	215.54	0.39%
R110	451.35	1.63%	C206	215.54	0.39%
R111	448.13	4.51%	C207	215.54	0.49%
R112	411.90	4.81%	C208	215.37	0.41%
Total distance		19,239.00	Mean rel. Distance		3.39%

*: Optimal solutions given in [10].

For the neural network parameters optimization an evolutionary strategy is used. Genes in the chromosome are the net weights and bias. The objective function is the total distance calculated solving a problem or a set of problems. The search algorithm used in this work is the Differential Evolution strategy as defined in [11]. The parameters used for the evolution are $F = 0.7$ and $CR = 0.3$.

Algorithm's performance is high (see table I), especially in the cluster type problems. In most of cases it is able to reach *quasi*-optimal results.

The solution analysis involves large amounts of generated data. Hence, in this paper only a piece of analysis is shown in order to illustrate its operation. The instance was randomly chosen and it is the problem RC108. First step is the analysis of independence among input vector components. It is performed using the statistical package R [12]. Independence test based on empirical Copula process is applied and generates as a result a *dependogram* which is used to identify associated factors. As it was expected, factors with analogous definition as the centrality factor and centrality order are found to be high correlated.

TABLE II
MODEL DERIVATIVE WITH RESPECT INPUT FACTORS

Factor	Average Derivative	Mex. Derivative
TAF	-0.479	3.058
CTF	-0.441	2.782
CNFR	0.366	2.353
RLF	-0.351	2.228
DDF	-0.229	1.526
WT	0.165	1.409
ADF	0.149	0.951
CNF	0.142	0.917
RGR	0.131	0.877
CNO	0.116	1.125
RTG	0.113	0.729
AR	-0.110	0.708
IDF	0.107	0.761
CNOR	0.100	0.701
DDF	0.065	0.521
CTO	-0.059	0.975

The components of neural network gradient are sorted according to average derivate (see Table II). In this case, two factors related to time window constraints are found to be the most relevant. They are the TAF (10) and CTF (12). This is characteristic of a problem in which time windows are the most restrictive factor. These rules are such that nodes with earlier arriving time from routes and earlier closing times are the most prioritized. The three following rules prioritize routes far away from depot, low vehicle load and low degree nodes.

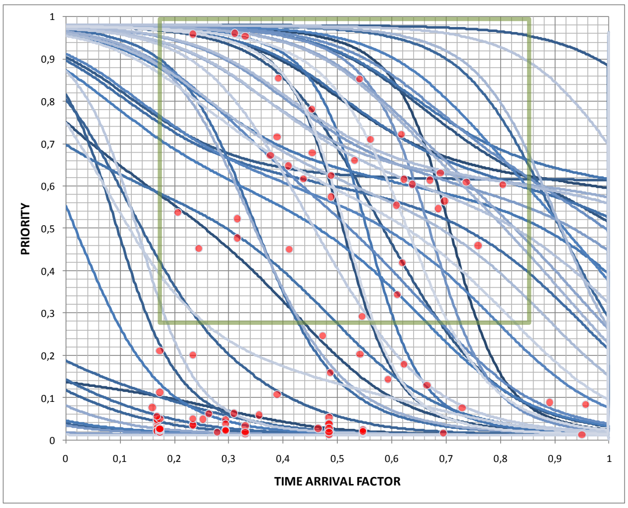

Fig. 1. Priority as a function of TAF generated by the ANN and distribution of values calculated during resolution of instance RC108. Green rectangle marks high priority assignations.

V. DISCUSSION AND CONCLUSION

A methodology for developing constructive heuristic algorithms for general routing problems has been introduced and tested on a well known problem as is the VRPTW. Algorithm's performance is good since its results are close to optimal solutions and reach the optimum on several instances

Two main lines of applicability derive from our methodology. First, it can be employed to solve different routing problems that arise when analyzing logistic systems. Second, the analysis of the obtained solutions allows the relevant factors for instances' resolution to be identified and thus the categorization of problems. Since the effect of problem's characteristics on the solution can be measured, it becomes possible to use this knowledge for the improvement of other algorithms or in the development of new ones.

But the most interesting contribution of our methodology is that it provides a systematic way for studying new routing problems for which little knowledge on suitable heuristic rules is available. This way, human effort is reduced and focused exclusively on the model selection together with its appropriate input factors as well as on the analysis of results.

Thus, other routing problems as the VRP or the PDP can also be tackled by simply replacing the characteristic problem factors and verifying specific constraints It can also be employed for the solution of dynamic routing problems. The solution procedure would start with an *a-priori* net's parameters tuning followed by an on line routing of the nodes using the obtained net.

Finally, a suitable set of techniques for an exploratory solution analysis has been illustrated. They lead to an enhanced comprehension of the evolved model's behavior resulting in a valuable learning from the solutions in order to define new algorithms.

ACKNOWLEDGMENT

We wish to express our gratitude to the Xunta de Galicia, which has funded this work through the research project "Análisis y desarrollo de una aplicación para la gestión, planificación y optimización de rutas de transporte en entornos dinámicos" (PGIDIT06DPI000PR10).

REFERENCES

[1] Miguel Andres Figliozzi, Hani S. Mahmassani, Patrick Jaillet, "Pricing in Dynamic Vehicle Routing Problems", *Transportation Science,* vol. 41, no. 3, pp. 302–318, 2007.

[2] Marius M. Solomon, and Jacques Desrosiers, "Time window constrained routing and scheduling problems", *Transportation Science,* vol. 22, no. 1, pp. 1–13, 1988.

[3] T. Van Woensel, L. Kerbache, H. Peremans, N. Vandaele, "Vehicle routing with dynamic travel times: A queueing approach", *European Journal of Operational Research,* vol. 186, pp. 990–1007, 2008.

[4] Chen Ai-ling, Yang Gen-ke and Wu Zhi-ming, "Time window constrained routing and scheduling problems", *Journal of Zhejiang University - Science A*, vol. 7, no. 4, pp. 607–614, 2005.

[5] R. Bai, J. Blazewicz, E.K. Burke, G. Kendall and B. McCollum, "A simulated annealing hyper-heuristic methodology for flexible decision support", Technical report, School of CSiT, University of Nottingham, UK. 2007.

[6] Gregory Gutin and Abraham P. Punnen (Eds.) "The travelling Salesman Problem and Its Variations", Kluwer Academic Publishers, 2002, ap. A, pp 750-753.

[7] Faulin, J., García del Valle, A., "Solving the capacitated vehicle routing problem using the ALGELECT electrostatic algorithm", Journal of the Operational Research Society, pp. 1-11, 2007.

[8] Hornik, K.; Stinchcombe, M. and White, H. "Multilayer feedforward networks are universal approximators", *Neural Networks*, no. 2, pp. 359-366, 1989.

[9] C. Genest and B. Rémillard, "Tests of independence and randomness based on the empirical copula process", *Test*, no. 13, pp. 335–369, 2004.

[10] Marius M. Solomon, http://web.cba.neu.edu/~msolomon/problems.htm, last updated March, 24, 2005. Last visited June, 2009.

[11] Rainer Dtorn and Kenneth Price "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *Journal of Global Optimization* no. 11, pp. 341–359, 1997.

[12] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing. Vienna, Austria, 2005.