Primal and Dual Neural Networks for Shortest-Path Routing

Jun Wang

Abstract—This paper presents two recurrent neural networks for solving the shortest path problem. Simplifying the architecture of a recurrent neural network based on the primal problem formulation, the first recurrent neural network called the primal routing network has even much simpler architecture. While being simple in architecture, the primal and dual routing networks are capable of shortest-path routing like their predecessor. Based on the dual problem formulation, the second recurrent neural network called the dual routing network has less complex connectivity than its predecessor. Therefore, parallel solution methods are more desirable.

Since Hopfield and Tank’s seminal work [8], [9], neural networks for solving optimization problems have been a major area in neural network research. While the mainstream of neural network approach to optimization focuses on solving the NP-complete problems such as the traveling salesman problem [8], there have been a few direct attacks of the shortest path problem using neural network [10]–[12]. These investigations have shown the sufficient potentials for the neural network approach to the shortest path problem.

In the present paper, two recurrent neural networks for shortest-path routing, called the primal and dual routing networks, are presented. These recurrent neural networks are capable of routing the shortest path in networks with mixed positive and negative cost coefficients. These recurrent neural networks are also much simpler in architecture, hence much easier for hardware implementation.

I. INTRODUCTION

The shortest path problem is concerned with finding the shortest path from a specified origin to a specified destination in a given network while minimizing the total cost associated with the path. The shortest path problem is an archetypal combinatorial optimization problem having widespread applications in a variety of settings. The applications of the shortest path problem include vehicle routing in transportation systems [1], traffic routing in communication networks [2], [3], and path planning in robotic systems [4]–[6]. Furthermore, the shortest path problem also has numerous variations such as the minimum weight problem, the quickest path problem, the most reliable path problem, and so on.

The shortest path problem has been investigated extensively. The well-known algorithms for solving the shortest path problem include the \( O(n^3) \) Bellman’s dynamic programming algorithm for directed acyclic networks, the \( O(n^2) \) Dijkstra-like labeling algorithm and the \( O(n^3) \) Bellman–Ford successive approximation algorithm for networks with nonnegative cost coefficients only, where \( n \) denotes the number of vertices in the network. See [7] for a comprehensive coverage of these algorithms. Besides the classical methods, many new and modified methods have been developed during the past few years. For large-scale and real-time applications such as traffic

Manuscript received June 27, 1996; revised December 7, 1997. This work was supported by the Hong Kong Research Grants Council under Grant CUHK38196E.

The author is with the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, Shatin, N.T., Hong Kong, R.O.C.

Publisher Item Identifier S 1083-4427(98)06216-X.

II. PATH REPRESENTATIONS

A. Problem Statement

Given a weighted direct graph \( G = (V, E) \) where \( V \) is a set of \( n \) vertices and \( E \) is an ordered set of \( m \) edges. A fixed cost \( c_{ij} \) is associated with the edge from vertices \( i \) to \( j \) in the graph \( G \). In a transportation or a robotic system, for example, the physical meaning of the cost can be the distance between the vertices, the time or energy needed for travel from one vertex to another. In a telecommunication system, the cost can be determined according to the transmission time and the link capacity from one vertex to another. In general, the cost coefficients matrix \( [c_{ij}] \) is not necessarily symmetric, i.e., the cost from vertices \( i \) to \( j \) may not be equal to the cost from vertices \( j \) to \( i \). Furthermore, the edges between some vertices may not exist, i.e., \( m \) may be less than \( n^2 \) (i.e., \( m < n^2 \)). The values of cost coefficients for the \( n^2 - m \) nonexistent edges are defined as infinity. More generally, a cost coefficient can be either positive or negative. A positive cost coefficient represents a loss, whereas a negative one represents a gain. It is admittedly more difficult to determine the shortest path for a network with mixed positive and negative cost coefficients [7]. For an obvious reason, we assume that there are neither negative cycles nor negative loops in the networks (i.e., \( \forall i, j \); \( c_{ii} \geq 0, \sum_{j=0}^{\infty} c_{ij} \geq 0 \)). Hence the total cost of the shortest path is bounded from below. Since the vertices in a network can be labeled arbitrarily, without loss of generality, we assume hereafter vertices 1 and \( n \) are origin and destination, respectively.

B. Vertex Path Representation

A path in a given network can be represented in different ways and the way of path representation in turn affects the effectiveness and efficiency of a solution procedure. In the literature, there are two typical path representations: vertex representation and edge representation.

Rauch and Winarske [10] and Lee and Chang [11] used an \( n \times p \) binary matrix to represent a path, where \( p \) is the number of vertices in the shortest path and which is assumed to be known. Specifically, the binary matrix of vertex path representation \( [x_{ij}] \) contains only

“0” and “1” elements, and $x_{ij}$ can be defined as

$$x_{ij} = \begin{cases} 1, & \text{if the } j\text{th vertex is the } i\text{th stop in the path;} \\ 0, & \text{otherwise.} \end{cases}$$  

(1)

Obviously, each row and each column in the binary matrix of vertex path representation contains no more than one “1” element, respectively, if each vertex can be visited no more than once. Since the binary matrix provides the path information in terms of vertices, this path representation is called a vertex representation. A limitation of the vertex path representation is that it requires determining the number of vertices in the shortest path $p$ a priori. In a dynamic environment, it is almost impossible to determine $p$ a priori due to the time-varying nature of cost coefficients. If $p$ is assigned with a number smaller than the number of vertices in the shortest path, then the resultant vertex representation cannot represent the shortest path and any solution procedure based on the representation cannot determine the shortest path.

C. Dual Formulation Based on Edge Path Representation

Based on the edge path representation, the primal shortest path problem can be formulated as a linear integer program as follows [14]:

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} \\
\text{subject to} & \quad \sum_{k=1}^{n} x_{ik} - \sum_{i=1}^{n} x_{ij} \\
& \quad = \begin{cases} 1, & \text{if } i = 1, \\ 0, & \text{if } i = 2, 3, \ldots, n - 1, \\ -1, & \text{if } i = n, \end{cases} \\
& \quad x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, n
\end{align*}$$

(9)

(10)

(11)

where $x_{ij}$ denotes the decision variable associated with the edge from vertices $i$ to $j$, as defined in (2). The objective function to be minimized, (9), is also the total cost for the path. The equality constraint coefficients and the right-hand sides are $-1$, $0$, or $1$. Equation (10) ensures that a continuous path starts from a specified origin and ends at a specified destination.

Because of the total unimodularity property of the constraint coefficient matrix defined in (10) [14], the integrality constraint in the shortest path problem formulation can be equivalently replaced with the non-negativity constraint, if the shortest path is unique. In other words, the optimal solutions of the equivalent linear programming problem are composed of zero and one integers if a unique optimum exists [14]. The equivalent linear programming problem based on the simplified edge path representation can be described as follows:

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n-1} \sum_{j=2, j \neq i}^{n} c_{ij}x_{ij} \\
\text{subject to} & \quad \sum_{k=1}^{n} x_{ik} - \sum_{i=1}^{n} x_{ij} \\
& \quad = \delta_{i1} - \delta_{in}, \quad i = 1, 2, \ldots, n \\
& \quad x_{ij} \geq 0, \quad i \neq j; \quad i = 1, 2, \ldots, n - 1 \\
& \quad j = 2, 3, \ldots, n
\end{align*}$$

(12)

(13)

(14)

where $\delta_{pq}$ is the Kronecker delta function defined as $\delta_{pq} = 1$ if $p = q$ and $\delta_{pq} = 0$ if $p \neq q$.

C. Dual Formulation Based on Edge Path Representation

Since the number of equality constraints $n$ is much less than the number of decision variables $(n - 1)(n - 2)$, it is more desirable to formulate and solve the dual of the primal shortest path problem. Based on the edge path representation, the dual shortest path problem
can be formulated as a linear programming problem as follows [14]:

\[
\begin{align*}
\text{maximize} & \quad y_n - y_1 \\
\text{subject to} & \quad y_j - y_i \leq c_{ij}, \quad i \neq j \\
& \quad i, j = 1, 2, \ldots, n
\end{align*}
\]  

(15)

where \( y_i \) denotes the dual decision variable associated with vertex \( i \).

Note that the value of the objective function at its maximum is the total cost of the shortest path [14]. Since the objective function and constraints in the dual problem involves variable differences only, an equivalent dual shortest path problem with \( n-1 \) variables can be formulated by defining \( z_i = y_i - y_1 \) for \( i = 1, 2, \ldots, n \):

\[
\begin{align*}
\text{maximize} & \quad z_n \\
\text{subject to} & \quad z_j - z_i \leq c_{ij}, \quad i \neq j; \\
& \quad i, j = 1, 2, \ldots, n
\end{align*}
\]

(16)

where \( z_1 \equiv 0 \). The value of the objective function at its maximum is still the total cost of the shortest path. Moreover, \( z_i \), \( i = 1, 2, \ldots, q \), is the cost of the shortest path with \( i \) edges to the destination where \( q \) is the number of edges in the shortest path. In addition, if \( z_i \) is used as the objective function to be maximized, then at optimum it is the shortest path from vertex 1 to vertex \( i \) [14].

Although the last component of the optimal dual solution gives the total cost of the shortest path, the optimal dual solution needs post-processing to decode the optimal primal solution in terms of edges. According to the Complementary Slackness Theorem [14]: given the feasible solutions of \( x_{ij} \) and \( z_i \) to the primal and dual problems, respectively, the solutions are optimal if and only if (1) \( x_{ij} = 1 \) implies \( z_j - z_i \equiv c_{ij} \) and (2) \( x_{ij} = 0 \) is implied by \( z_j - z_i < c_{ij} \) for \( i, j = 1, 2, \ldots, n \). The Complementary Slackness Theorem can be used as the basis for the post-processing as will be discussed in Section V.

The shortest path problem formulation based on the edge path representation, (12)–(14) or (17) and (18), is a linear programming problem, whereas the problem based on the vertex path representation, (3)–(8), is an integer nonconvex quadratic programming problem. The advantages of the edge representation over the vertex representation become more obvious. The subsequent development of this paper is thus based on the simplified edge path representation.

IV. PRIMAL ROUTING NETWORK

Because the shortest path problem based on the edge path representation is formulated as a linear program, it can be solved by the neural networks proposed for linear programming. In [15], [16], a recurrent neural network called the deterministic annealing network is presented and demonstrated to be capable of solving linear programming problems. The primal and dual routing networks are tailored from the deterministic annealing network [15], [16]. Let the decision variables of the primal and dual shortest path be represented, respectively, by the activation states of the primal and dual routing networks. For simplicity of notations, the same symbols \( [x_{ij}] \) and \( [z_i] \) are used to denote both the decision variables and corresponding activation states.

A. Energy Function

An energy function for the primal problem based on the simplified edge path representation can be defined as follows:

\[
E_p[t,x(t)] = \frac{w^2}{2} \sum_{i=1}^{n} \left\{ \sum_{j=1,j\neq i}^{n} \left[ x_{ij}(t) - x_{i1}(t) \right] - \delta_{1i} + \delta_{in} \right\}^2 + \alpha \sum_{i=1}^{n-1} \sum_{j=2,j\neq i}^{n} c_{ij} \exp(-t/\tau)x_{ij}(t)
\]

(19)

where \( w, \beta, \) and \( \tau \) are positive scaling constants and \( \alpha \exp(-t/\tau) \) is a decaying temperature parameter. The role of the temperature parameter in (19) is explained at length in [26] and [27].

B. Dynamical Equation

Let \( du_{ij}(t)/dt = -\partial E_p[t,x(t)]/\partial x_{ij} \), based on the simplified edge path representation, the state dynamical equation and the output function of the recurrent neural network presented in [13] is as follows: for \( i \neq j = 1, 2, \ldots, n-1; j = 2, 3, \ldots, n; \)

\[
\frac{du_{ij}(t)}{dt} = -w \sum_{k=2,k\neq j}^{n} x_{ik}(t) + w \sum_{l=1,l\neq i}^{n-1} x_{il}(t)
\]

\[
+ w \sum_{p=2,p\neq j}^{n} x_{ip}(t) - w \sum_{q=1,q\neq j}^{n-1} x_{iq}(t)
\]

\[
+ w(\delta_{1j} - \delta_{1i} - \delta_{jn} + \delta_{in}) - \alpha c_{ij} \exp(-t/\tau)
\]

(20)

\[
x_{ij}(t) = f[u_{ij}(t)]
\]

(21)

where \( u_{ij}(t) \) denotes the net input to neuron \( (i, j) \), \( f(u) \) is a nonnegative and nondecreasing activation function defined as \( f(u) \geq 0 \) if \( u \geq 0 \) or \( f(u) = 0 \) otherwise, and \( df(u)/du \geq 0 \).

To reduce the complexity of the resulting neural network architecture, the dynamical equation and output function of the present primal routing network can be described by using \( n \) instrumental variables \( v_i(t), i = 1, 2, \ldots, n \):

\[
\frac{dv_i(t)}{dt} = -wv_i(t) + wv_{j}(t) - \alpha c_{ij} \exp(-t/\tau)
\]

(22)

\[
v_i(t) = \sum_{k=2,k\neq i}^{n} x_{ik}(t) - \sum_{l=1, l\neq i}^{n-1} x_{il}(t)
\]

\[
- \delta_{1i} + \delta_{in}
\]

(23)

\[
x_{ij}(t) = f[v_i(t)]
\]

(24)

C. Network Architecture

The primal routing network consists of \( n^2 - 2n + 2 \) neurons arranged spatially in two layers: an output layer and a hidden layer. The output layer consists of an \((n-1) \times (n-1)\) two-dimensional array of output neurons without the diagonal elements representing \( [x_{ij}] \). The hidden layer consists of an \( n \)-vector of hidden neurons representing instrumental variables \( [v_i] \).

The first two terms in the right-hand side of (22) define the connectivity from the \((n-1)\) hidden neurons to the \((n-1)\) output neurons. The third term in the right-hand side of (22) defines a decaying external input \( \alpha \exp(-t/\tau)c_{ij} \) to the output layer. Similarly, the first two terms in the right-hand side of (23) define the connectivity from the \((n-1)\) output neurons to the \( n \) hidden neurons. The third term in the right-hand side of (23) defines a constant positive and negative unity input (bias) to the hidden neurons corresponding, respectively, to the specified origin and destination. Moreover, (22) and (23) also show the connectivity of the primal routing network:

1) there exist \((n-1)(n-2)\) excitatory connections with weight of \( w \) to \( x_{ij}(t) \) in the output layer from \( v_i(t) \) in the hidden layer;

2) there exist \((n-1)(n-2)\) inhibitory connections with weight of \(-w\) to \( x_{ij}(t) \) in the output layer from \( v_i(t) \) in the hidden layers;
From (22), it is easy to see that the convergence rate of primal routing network depends also on the parameter $\tau$. The parameter $\tau$ serves as a time constant for the decaying bias to reinforce the effect of cost minimization. The time constant $\tau$ has to be sufficiently large to sustain cost minimization and ensure solution optimality. Since the transient time of primal routing network depends on the reciprocals of the sensitivity parameter $\xi$ and the minimum absolute value of the nonzero eigenvalues of the connection weight matrix $nw$, a lower bound on the value of $\tau$ is $4/(\xi nw x_{\text{max}})$, i.e., $\tau > 4/(\xi nw x_{\text{max}})$ if the unipolar sigmoid activation function is used, as discussed in [15].

Based on a similar analysis, a design rule is that $\tau > 1/(\xi nw)$ if the Heaviside activation function is used. The role of $\alpha$ is to balance the effects of constraint satisfaction and cost minimization. Let $c_{\text{max}} = \max\{c_{ij}; i, j = 1, 2, \ldots, n\} < \infty$. A design rule is to select $\beta$ such that $\beta \geq w/c_{\text{max}}$.

V. DUAL ROUTING NETWORK

A. Energy Functions

Similarly to the primal shortest path problem, an energy function for the dual problem can be formulated as

$$E_d[t, z(t)] = \frac{w}{\tau} \sum_{i=1}^{n} \sum_{j \neq i} [g[z_j(t) - z_i(t) - c_{ij}]]^2$$

$$- \beta \exp(-t/\tau) z_n(t)$$

(25)

where $w, \beta, \tau > 0$, $g(\cdot)$ is a nonnegative and nondecreasing activation function defined as $g(z) > 0$ if $z > 0$ or $g(z) = 0$, otherwise, and $\beta \exp(-t/\tau)$ is a decaying temperature parameter.

B. Dynamical Equation

Let $d z_i(t)/dt = -\partial E_d[t, z(t)]/\partial z_i$, the dynamical equation and output function of the dual routing network are as follows:

$$\frac{dz_i(t)}{dt} = -w \sum_{j \neq i} [g[z_j(t) - z_i(t) - c_{ij}]]$$

$$+ \delta_{ij} \beta \exp(-t/\tau), \quad i = 2, 3, \ldots, n$$

$$x_{ij}(t) = h[z_j(t) - z_i(t) - c_{ij}]$$

(26)

where $h(u)$ is the output function defined as $h(u) = 1$ if $u = 0$, or $h(u) = 0$ otherwise.

C. Network Architectures

The dual routing network consists of $n - 1$ neurons representing $z_i$ arranged spatially in a layer. The dynamical equation (26) of the dual routing network shows that there is an inhibitory connection with weight of $-w$ and an excitatory connection with weight of $w$ from every pair of $z_i(t) (i = 2, 3, \ldots, n)$. That is, the number of connections is $4(n-1)^2$, the same as the primal routing network. The dynamical equation also shows that only the neuron corresponding to the destination has a decaying external input $\beta \exp(-t/\tau)$. Fig. 2 illustrates the architecture of the dual routing network.
D. Design Parameters

Because the constraint coefficient matrix in a dual problem is the transpose of that in the primal problem, it can be seen that the nonzero eigenvalues of the system matrix of the linearized dual routing network are also $-2nw$. Large $w$ can expedite the convergence of the dual routing network as well. Similar to $\alpha$ in the primal routing network, the role of $\beta$ is to balance the effects of constraint satisfaction and objective maximization. It is usually set $\beta \approx w^\text{max}$.

Similar to that in the primal routing network, the role of the activation function in the dual routing network is to enforce the inequality constraints (18) and scale the sensitivity of the activation. The Heaviside activation function discussed in the preceding section is suitable for the dual routing network.

E. Convergence Analysis

The asymptotic stability of the deterministic annealing network is proven in [16]. The necessary and sufficient condition for the state variables of the dual routing network to converge to a feasible solution will be shown to be the absence of negative cycles in the given network $G$. A negative cycle is characterized by the negative sum of cost coefficients around a closed circuit in a network, i.e., $\exists j$ such that $c_{ij} + c_{jk} + \cdots + c_{il} < 0$. To prove the necessity, adding both sides of the constraints associated with indices $i, j, k, \ldots$, $z_j - z_i \leq c_{ij}, z_j - z_k \leq c_{jk}, \ldots, z_l - z_i \leq c_{il}$ results in $c_{ij} + c_{jk} + \cdots + c_{il} \leq 0$. The sufficiency can be proved by showing that the convergence to an infeasible solution will lead to the presence of at least one negative cycle. Examining the dynamical equation (26) of the dual routing network, we can see that the only cause of an infeasibility convergence results from the existence of an offset effect in the first term on the right-hand side, since the second (last) term vanishes as time approaches infinity and the left-side hand approaches zero as the states converge. Specifically, an infeasibility implies that at least one term in both $\sum_{j \neq i} g[z_j(t) - z_i(t) - c_{ij}]$ and $\sum_{j \neq l} g[z_j(t) - z_l(t) - c_{lj}]$ are positive. According to the definition of $g(\cdot)$ in the dual routing network, $g[z_j(\infty) - z_i(\infty) - c_{ij}] > 0$ if and only if $z_j(\infty) - z_i(\infty) - c_{ij} > 0$. If the cause of infeasibility is that $g[z_j(\infty) - z_i(\infty) - c_{ij}] > 0$ and $g[z_j(\infty) - z_k(\infty) - c_{ik}] > 0$, then $z_j(\infty) - z_k(\infty) - c_{ik} > 0$. Adding both sides of the two inequalities results in a negative cycle $c_{ij} + c_{ik} < 0$. If an infeasibility is caused by $z_l(\infty) - z_j(\infty) - c_{lj} > 0$ and $z_k(\infty) - z_l(\infty) - c_{lk} > 0$, implies $3m$ such that $z_m(\infty) - z_j(\infty) - c_{mj} > 0$ and $z_k(\infty) - z_l(\infty) - c_{lk} > 0$. These inequalities in turn result in two or other inequalities. Eventually, adding both sides of all the resulting inequalities lead to the presence of a negative cycle $c_{ij} + c_{mk} + \cdots + c_{nk} + c_{kl} < 0$. The case that an infeasibility results from the cancellation of more than two terms can be proven in the similar way.

F. Post Processing

Using the Complementary Slackness Theorem via the output function (27), the optimal primal solution in terms of edges can be decoded from the optimal dual solution. The activation states from the output function of the dual routing network, however, sometimes results in an infeasible primal solution (specifically, more “1” than required). This infeasibility can be easily removed by checking every element of the resulting primal solution matrix $[z]$ from origin to destination and converting inconsistent “1” to “0.” Specifically, $\forall x_{ij} = 1$, if $\forall k, x_{kj} = x_{jk} = 0$ then set $x_{ij} = 0$.

VI. ILLUSTRATIVE EXAMPLES

Example 1: Consider the shortest path problem with 10 vertices (Example 1 in [13] or Example 2 in [17]) where the origin and destination vertices are, respectively, vertices 1 and 10. The Euclidean distances are used as the cost coefficients. The shortest path of this problem is $e_{12}, e_{23}, e_{310}$ where $e_{ij}$ ($i = 1, 2, \cdots, n$) denotes the edge between vertices $i$ and $j$ in the network. The total cost of the shortest path is 1.149896. Let $w = \beta = 10^5, \tau = 10^{-7}$, and $c_{\infty} = 10$. Fig. 3 depicts the transient behavior of the activation states of the dual routing network. The steady-state vector of the output neurons is [0.325 0.914, 0.510 0.966, 0.585 0.029, 1.207 0.522, 0.476 0.889, 0.701 0.207, 0.555 0.255, 0.831 0.225, 1.149 0.902]. After post-processing to remove inconsistency and ensure feasibility, the neural network solution to the problem represents the shortest path. The simulated recurrent neural network takes about 0.5 ms to converge.

Unlike the popular Dijkstra’s labeling algorithm which can solve the shortest path problem with nonnegative cost coefficients only, the primal and dual routing networks are capable of solving the shortest path problem with mixed-sign cost coefficients as will be shown in the following example.

Example 2: Consider the 10-vertex directed network with mixed positive and negative cost coefficients (Example 2 in [13]). The cost coefficient matrix is asymmetric and there are no loops or negative cycles in this network. The shortest path of this problem is $\{e_{19}, e_{92}, e_{25}, e_{57}, e_{7, 10}\}$, with the total cost of 0.360 952. Fig. 4 depicts the transient behavior of the activation states of the dual routing network simulated using the same design parameters in Example 1. The steady-state vector of the output neurons is [0.053 0.502, 0.041 0.353, 0.268 0.410, 0.037 0.663, 0.345 0.777, 0.212 0.687, 0.211 0.472, 0.117 0.621, 0.360 0.951]. After post-processing to remove inconsistency and ensure feasibility, the neural network solution to the problem represents the shortest path. It takes less than 0.5 ms (5\tau) for the simulated dual routing network to converge.

VII. CONCLUDING REMARKS

In this paper, the primal routing network with $O(n^2)$ neurons and connections and the dual routing network with $O(n)$ neurons and $O(n^2)$ connections for solving the shortest path problem have been presented. The dynamics and architectures of the primal and dual routing networks have been described. The design methodology and guidelines have been delineated. It has been shown that the primal and dual routing networks are capable of shortest-path routing for directed
networks with arbitrary cost coefficients. The convergence rate of the primal and dual routing networks is nondecreasing with respect to the size of the shortest path problem and can be expedited by properly scaling design parameters. These features make the primal and dual routing networks suitable for solving large-scale shortest path problems in real-time applications. One salient advantage of the primal and dual routing networks is the independence of the connection weight matrix upon specific problems. Specifically, only the constant biases are different for different origins and/or destinations of the same network, and only the initial values of the decaying biases are different for different networks with the same number of vertices, and the same origin and destination. By energizing different vertices, the primal and dual routing networks can be used to generate all-pair shortest paths and minimal spanning trees. These desirable features facilitate the VLSI implementation of the primal and dual routing networks. The primal and dual routing networks implemented in a VLSI circuit can serve as coprocessors for onboard planning in dynamic decision environments.

REFERENCES