2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

A Discrete Harmonic Potential Approach to Motion Planning on A Weighted Graph

Ahmad A. Masoud
Electrical Engineering Department, KFUPM, P.O. Box 287, Dhaharan 31261, Saudi Arabia, masoud@kfupm.edu.sa

***Abstract*-**A provably-correct discrete version of the harmonic potential field (HPF) approach to motion planning was suggested in [20]. The approach utilizes the strong relation between graph theory and electrical network theory for developing a framework of theories and definitions that, among other things, can strongly aid in developing a discrete HPF planning approach. This framework was used to suggest an efficient, optimal, novel, discrete planning method called the M* algorithm. In this paper an in-place, successive relaxation procedure is suggested for implementing the M* algorithm. Also, the utility of the discrete HPF approach is demonstrated in robust, data network routing.

I. Introduction

Harmonic potential fields (HPFs) provide a means by which the behavior of a robot may be sensitized in a provably-correct, constrained, goal-oriented manner to the context in which a robot is operating. An HPF is generated using a Laplace boundary value problem (BVP) configured using a properly chosen set of boundary conditions. There are several settings one may use for a Laplce BVP (LBVP) in order to generate a navigation potential [1]. Each one of these settings possesses its own, distinct, topological properties [2]. An example is shown below of an LBVP that is configured using the homogeneous Neumann boundary conditions:

$$\nabla^2 V(X) \equiv 0 \qquad X \in \Omega \qquad (1)$$

subject to: $V(X_S) = 1$, $V(X_T) = 0$, and $\frac{\partial V}{\partial \mathbf{n}} = 0$ at $X = \Gamma$,

where Ω is the workspace, Γ is its boundary, $\mathbf{n}$ is a unit vector normal to Γ, X_s is the start point, and X_T is the target point. Harmonic functions have many useful properties [3,4] for motion planning. Most notably, a harmonic potential is also a Morse function and a general form of the navigation function suggested in [5] (see [19]). The HPF approach may be configured to operate in a model-based and/or sensor-based mode. It can also be made to accommodate a variety of differential and state constraints [6]. An HPF planner could be used to generate the kinematic guidance signal only, or it may be used to directly generate the navigation control signal for a holonomic system with second order dynamics [7]. An HPF may also be realized as a large scale, parallel-distributed machine with simple, locally-connected processing nodes [17].

This paper contributes a theoretical framework that may be used for providing a discrete counterpart to the HPF approach. The framework can serve either as a basis for constructing provably-correct procedures for synthesizing motion on a weighted graph or for augmenting the capabilities of existing algorithms of such a sort. This was demonstrated by suggesting a novel and efficient algorithm for finding the optimal path on a weighted graph. The algorithm does not require backtracking, and may be implemented in-place (as shown in this paper) eliminating the need for excessive storage. The framework is also used to suggest a generic solution to the lower bound problem encountered by the A* algorithm [18].

The framework is expected to assist in controlling the growth of the computational effort experienced by the HPF approach when planning is to be carried out in high dimensional spaces. Some researchers used a harmonic potential for biasing sampling planners [9,10] for this purpose. While gains in terms of improving the resolution were achieved, this may not be needed since the HPF approach has the ability to operate, on its own, in a sampling mode. An attempt based on Green's functions was made in [8] to configure the HPF approach to work in a sample-based mode; however, several difficulties were encountered. The most serious ones are: limitation of the approach to working in at most six degrees of freedom space, something that is well within the capabilities of the continuous version of the HPF. Moreover, the method in [8] is a heuristic method with no guarantees that if a solution exist it can be found.

The framework is also expected to contribute to the area of robust data packet transfer in a network of routers. The discrete HPF paradigm seems to support routing on-the-fly while the network is still in a transient state. Having the number of routers and the connectivity structure fixed is a perquisite for the discrete potential field to converge. This in turns makes it possible to derive an optimal routing action. Although optimality is desirable, service availability is a core requirement that needs to be maintained under abnormal conditions such as quick change of network connectivity, phasing-in or phasing out of router nodes. It is demonstrated that if a path to the target always exist and the switching delays in the routers are negligible, the packet will reach its destination despite the changes in the network which may simultaneously take place while the packet is being routed. This feature enables the construction of a two-tier motion controller for the network that consists of a single module capable of switching, in a transparent manner, between a centralized optimal mode and a ground-state, decentralized one that become active when the centralized mode can no longer be sustained.

The transition from the continuous, HPF case to the discrete one is made possible by the strong relation graph theory has to electrical networks [11,12,13]. It is shown in this paper that a discrete counterpart of the BVP in (1) may be established by replacing the Laplace operator with the flow balance operator represented by Krichhoff current law (KCL) [14]. As for the boundary conditions, they are applied in the same manner as in (1) to the boundary vertices.

In section II a discrete counterpart to (1) along with basic propositions are provided. In section III the M* algorithm is presented and in IV a lower bound for the A* is proposed. Section V contains a procedure that is based on the discrete HPF approach for routing on-the-fly. Conclusions are placed in section VI.

II. Propositions and definitions

This section provides basic propositions and definitions that are needed for constructing the minimum path algorithms.

Definition -1: Let G be a non-directed graph containing N vertices. Let the cost of moving from vertex i to vertex j be C_{ij} (C_{ij}=C_{ji}). Let a potential V_i be defined at each vertex of the graph (i=1,..,N), and I_{ij} be the flow from vertex i to vertex j defined as:

$$I_{ij} = \frac{V_i - V_j}{C_{ij}} \quad , \tag{2}$$

where $V_i > V_j$ .Note that equation-2 is analogous to ohm's law in electric circuits [14]. Let T and S be the target and start boundary vertices respectively.

A discrete counterpart for the BVP in (1) is obtained if at each vertex of G (excluding the boundary vertices) the balance condition represented by KCL is enforced:

$$\sum_j I_{ij} = 0 \qquad i=1,..,N,\ i \neq T,\ i \neq S$$

and $$V_S = 1,\ V_T = 0. \tag{3}$$

Definition-2: Let the equivalent cost between any two arbitrarily chosen vertices, i and j, of G (Ceq_{ij}) be defined as the potential difference applied to the i-j port of G (ΔV) divided by the flow, I, entering vertex i and leaving vertex j (figure-1)

$$Ceq_{ij} = \frac{\Delta V}{I} \tag{4}$$

Proposition-1: The equivalent cost of a graph, G, that satisfies KCL at all of its nodes (i.e. an electric network) as seen from the i-j port (vertices) is less than or equal to the sum of all the costs along any forward path connecting vertex i to vertex j. Note that if the proposition holds for forward paths, it will also hold for paths with cycles.

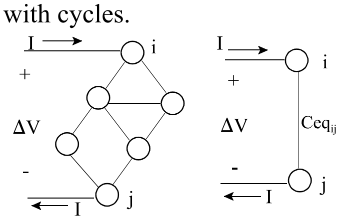


Figure-1: Equivalent cost of a graph as seen from the i-j vertices.

Proof: see [20].

Proposition-2: If G satisfies the conditions in equation-3, then the potential defined on the graph (V(G)) will have a unique minimum at T (V_T) and a unique maximum at S (V_S).
Proof: see [20].

Proposition-3: Traversing a positive, outgoing flow from any vertex in G will generate a sequence of vertices (i.e. a path) that terminates at T. Vice versa, traversing a negative, ingoing flow from any vertex in G will generate a sequence of vertices (i.e. a path) that terminates at S.
Proof: see [20]

Proposition-4: A path linking S to T generated by moving from a vertex to another using a positive flow cannot have repeated vertices (i.e. it contains no loops).
Proof: see [20].

Definition-3: Since a positive flow path (PFP) beginning at S is guaranteed to terminate at T with no repeated vertices in-between, the combination of all PFPs define a tree with S as the top parent vertex and bottom, offspring vertices equal to T. This tree is called the harmonic flow tree (HFT).

Proposition-5: The HFT of a graph contains all the vertices in that graph.
Proof: see [20]

Proposition-6: The HFT of a graph contains the optimal path linking S to T.
Proof: see [20]

Proposition-7: The optimum path (or any PFP for that matter) must contain at most N vertices.
Proof: see [20].

III. The **M*** Algorithm

In this section an algorithm (**M***) is suggested for computing the optimum path between S and T on a graph:

01. Write the KCL equations for each vertex of the graph

$$\sum_j I_{ij} = 0 \quad i = 1,...,N \tag{5}$$

02. From the KCL equations derive the vertex update equations:

$$V_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} b_{i,k} V_k \tag{6}$$

03. Initialize the variables:

$$V_S=1;\ V_T=0;\ V_i=1/2 \quad i=1,..,N \quad i \neq S,\ \ i \neq T \tag{7}$$

04. Loop till convergence is achieved performing:

$$V_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} b_{i,k} V_k \quad i=1,..,N,\ i \neq S,\ \ i \neq T \tag{8}$$

05. Compute the flows
06. Using the flows construct the HFT of the graph
07. Starting from the last parent nodes, for each node retain the branch with lowest cost and delete the others
08. Move to the parent nodes one level up and repeat step 7.
09. Repeat step 8 till the top parent node S is reached
10. The remaining branch connected to S is the optimal path linking S to T.

A. *An Example*

Consider the weighted graph shown in figure-2. It is required that a minimum cost path be found from the start vertex S=1 to the target vertex T=5. The transition costs are: C_{16}=1, C_{14}=3, C_{23}=4, C_{34}=7, C_{26}=1, C_{37}=5, C_{35}=2, C_{47}=6, C_{67}=9, C_{57}=5. The update equations are:

$$V_1 = \frac{1}{K_1}[\frac{V_4}{C_{14}} + \frac{V_6}{C_{16}}] = b_{1,4}V_4 + b_{1,6}V_6 \tag{9}$$
$$V_2 = \frac{1}{K_2}[\frac{V_3}{C_{23}} + \frac{V_6}{C_{26}}] = b_{2,3}V_3 + b_{2,6}V_6$$
$$V_3 = \frac{1}{K_3}[\frac{V_2}{C_{23}} + \frac{V_4}{C_{34}} + \frac{V_5}{C_{35}} + \frac{V_7}{C_{37}}] = b_{3,2}V_2 + b_{3,4}V_4 + b_{3,5}V_5 + b_{3,7}V_7$$
$$V_4 = \frac{1}{K_4}[\frac{V_1}{C_{14}} + \frac{V_3}{C_{34}} + \frac{V_7}{C_{47}}] = b_{4,1}V_1 + b_{4,3}V_3 + b_{4,7}V_7$$
$$V_5 = \frac{1}{K_5}[\frac{V_3}{C_{35}} + \frac{V_7}{C_{57}}] = b_{5,3}V_3 + b_{5,7}V_7$$
$$V_6 = \frac{1}{K_6}[\frac{V_1}{C_{1,6}} + \frac{V_2}{C_{2,6}} + \frac{V_7}{C_{6,7}}] = b_{6,1}V_1 + b_{6,2}V_2 + b_{6,7}V_7$$
$$V_7 = \frac{1}{K_7}[\frac{V_3}{C_{37}} + \frac{V_4}{C_{47}} + \frac{V_5}{C_{57}} + \frac{V_6}{C_{67}}] = b_{7,3}V_3 + b_{7,4}V_4 + b_{7,5}V_5 + b_{7,6}V_6,$$

where

$$\frac{1}{K_1} = [\frac{1}{C_{14}} + \frac{1}{C_{16}}] \qquad \frac{1}{K_2} = [\frac{1}{C_{23}} + \frac{1}{C_{26}}] \qquad \frac{1}{K_3} = [\frac{1}{C_{23}} + \frac{1}{C_{34}} + \frac{1}{C_{35}} + \frac{1}{C_{37}}] \qquad \frac{1}{K_4} = [\frac{1}{C_{14}} + \frac{1}{C_{34}} + \frac{1}{C_{47}}]$$
$$\frac{1}{K_5} = [\frac{1}{C_{35}} + \frac{1}{C_{57}}] \qquad \frac{1}{K_6} = [\frac{V_1}{C_{16}} + \frac{V_2}{C_{26}} + \frac{V_7}{C_{67}}] \qquad \frac{1}{K_7} = [\frac{1}{C_{37}} + \frac{1}{C_{47}} + \frac{1}{C_{57}} + \frac{1}{C_{67}}],$$

Setting $V_1=1$, $V_5=0$ and applying the procedure described above we obtain the vertices potential: $V_1=1$, $V_2=0.74673$, $V_3=0.33753$, $V_4=0.70006$, $V_5=0$, $V_6=0.84902$, $V_7=0.41093$. The flows may be computed as (figure-4): $I_{14}=0.09998$, $I_{16}=0.15098$, $I_{23}=0.1023$, $I_{47}=0.048189$, $I_{43}=0.05179$, $I_{35}=0.16877$, $I_{62}=0.1023$, $I_{67}=0.048677$, $I_{73}=0.014679$, $I_{75}=0.082186$

As can be seen (figure-4) the optimum path: 5→ 3→2→6→1 with a cost 8 was obtained after only two levels of branch removal.

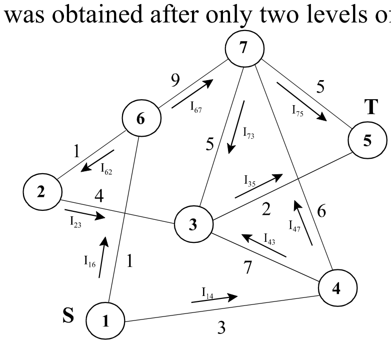


Figure-2: The graph and the flows.

Now construct the corresponding HFT (figure-3).

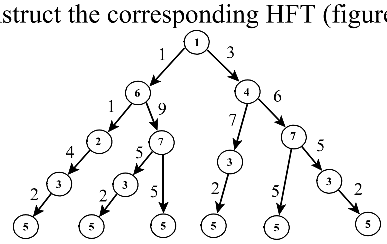


Figure-3: The HFT.

Start successively removing branches (figure-4):

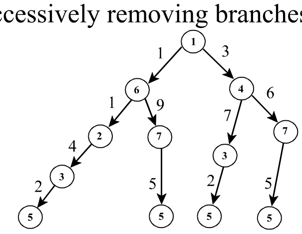


Figure-4a: Level-1 parent node branch removal.

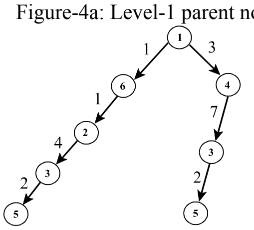


Figure-4b: Level-2 parent node branch removal.

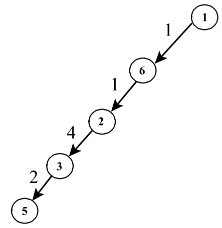


Figure-4c: Level-3 parent node branch removal .

B. *A successive relaxation procedure for implementing M**

The rapid growth of an HFT with the size of a graph makes it impractical to apply the algorithm on the tree directly. Here a procedure that makes it possible to operate on the HFT indirectly by successively relaxing the graph. In order to apply the procedure the following terms need to be defined (figure-5): positive flow index (PFI) of a vertex: number of edges connected to the vertex with outward positive flows. Negative flow index (NFI) of a vertex: is the number of edges connected to the vertex with inward negative flows.

Vertex				
PFI	0	1	2	3
NFI	3	2	1	0

Figure-5: PFI and NFI of a vertex.

The procedure is:

0-compute the PFI and NFI for each vertex of the graph,

1-starting from the target vertex and using the negative flows along with the NFIs and PFIs of the vertices, detect the junction vertices and label them based on their levels,

2- at the encountered junction vertex clear NB buffers where NB=PFI of the junction vertex,

3- now starting from the junction vertex, trace forward all paths to the target vertex traversing vertices with positive flows and PFIs=1,

4- excluding the lowest cost path, delete all the edges in the graph connecting the junction vertex to the first, subsequent vertices in the remaining paths,

5- decrement the PFI of the junction vertex by the number of edges removed from the graph,

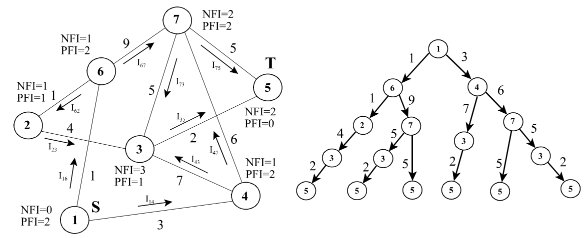
6- decrement the NFIs of the first subsequent vertices from step 4 by 1,

7- if the NFI of any vertex in the graph from step 4 is equal zero and the vertex is not a start vertex, delete the edge in the graph connecting that vertex to the subsequent vertex and reduce the NFI of the subsequent vertex by 1,

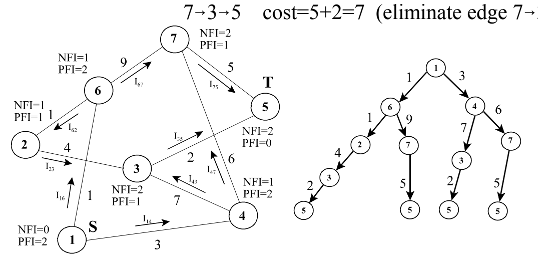
8- repeat 7 till all the reaming vertices in the paths with PFIs=1 have NFIs >0,

9- go to 1 and repeat till there is only one branch left in the graph with two terminal vertices having NFI=0, PFI=1 and NFI=1, PFI=0. This is the optimum path connecting the start vertex to the target vertex.

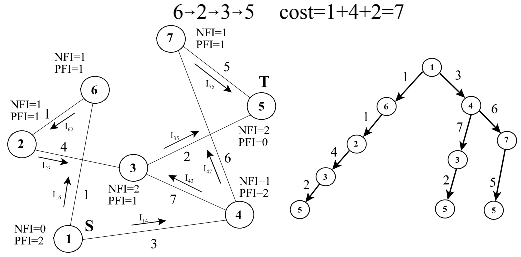
In the following the procedure is applied to the graph in figure-2 in a step by step manner (figure-6):



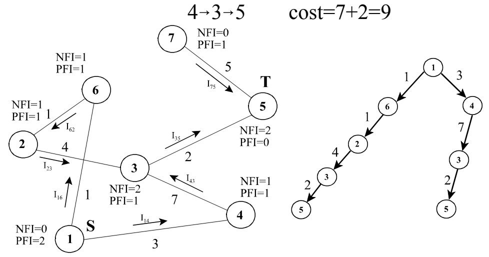
initially traced path: 5→7
constructed paths: 7→5 cost=5
7→3→5 cost=5+2=7 (eliminate edge 7→3 in graph)



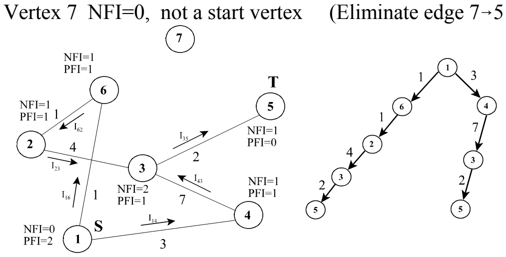
initially traced path: 5→3→2→6
constructed paths: 6→7→5 cost=9+5=14 (eliminate edge 6→7 in graph)
6→2→3→5 cost=1+4+2=7



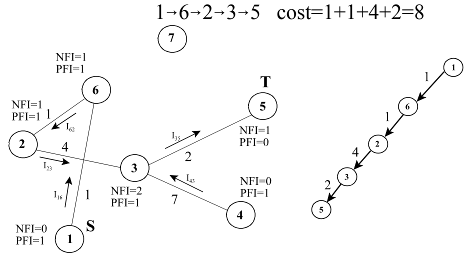
initially traced path: 5→7→4
constructed paths: 4→7→5 cost=6+5=11 (eliminate edge 4→7 in graph)
4→3→5 cost=7+2=9



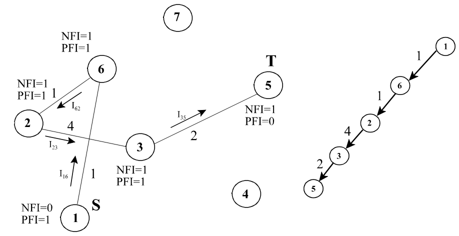
Vertex 7 NFI=0, not a start vertex (Eliminate edge 7→5 in graph)



initially traced path: 5→3→2→6→1
constructed paths: 1→4→3→5 cost= 3+7+2=12 (eliminate edge 1→4 in graph)
1→6→2→3→5 cost=1+1+4+2=8



Vertex 4 has an NFI=0 and is not a start vertex (eliminate edge 4 → 3)



All intermediate vertices have an NFI=PFI=1 → Algorithm terminates.

Figure-6: successive relaxation for the M*

The optimum path is: 1→6→2→3→5 having a cost of 8.

IV. A Lower bound for A*

For the A* algorithm to work a lower bound on the cost from each vertex of the graph to the target vertex has to be supplied. For spatial planning problems the Euclidian distance between the vertices provides such a bound. However, for the general case finding a lower bound may be a source of difficulties that prevents the use of the A* algorithm. In the followings it is shown that the concept of equivalent cost (resistance) from the resistive network paradigm can effectively solve this problem.

Consider the simple graph in figure-7 where S=1 and T=4. To apply the A* algorithm, the path at node 1 should be expanded towards 2 and 3. In order to sort the paths so that the next path expansion can be determined, lower estimates on the cost of moving from 2 to 4 and 3 to 4 are needed. Expansion of the path towards 2 may be achieved by simply removing all the edges of the graph that are attached to 1 leaving only the edge connected to vertex 2 (figure-7). The flows are then computed for the remaining part of the network. The equivalent cost from 2 to 4 (Ceq_{24}) may be computed as:

$$Ceq_{24} = \frac{1}{I_{12}} - C_{12}. \qquad (10)$$

Since in proposition-1 it is proven that the equivalent cost between two vertices in a graph is less than or equal to the least cost path connecting these vertices, the equivalent cost may be used as the lower bound estimate required by the A* algorithm. The minimum cost bounds needed for the remaining path expansions may be obtained in a similar manner.

A. *Example:*

The same example in the previous section is repeated using the A* algorithm and the equivalent cost concept. The successive path expansions are shown in figure-7. The optimum path is: 1→6→2→3→5 having a cost of 8.

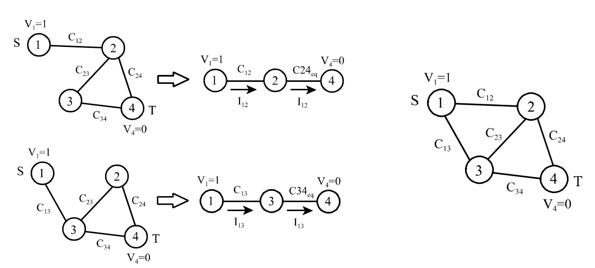


Figure-7: Path expansion

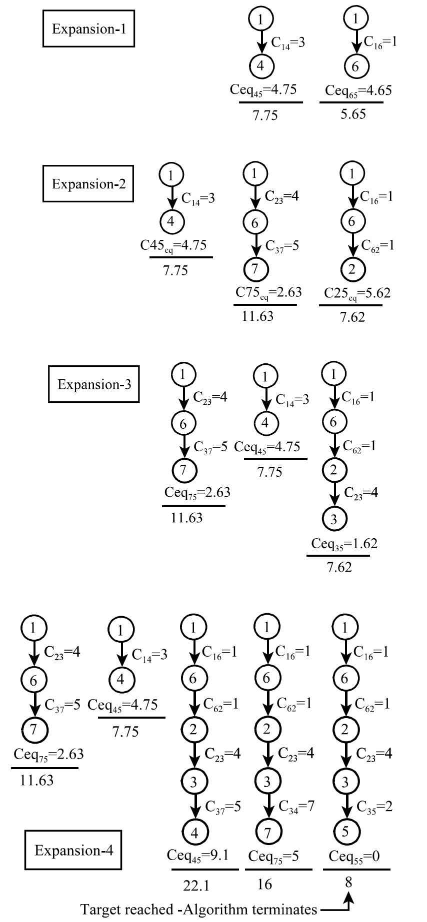


Figure-8: A* applied to the graph in figure-4.

V. Routing on-the-fly

In the previous sections optimal algorithms for planning motion on a weighted graph utilizing the flow in a resistive grid are suggested. In order to apply these algorithms the graph must have a fixed structure known to the central unit that is processing the data and generating the path. While the above setting applies in many practical situations there are cases where such a scenario cannot be applied, e.g. ad-hoc networks. Also, reliability and cost issues may make it undesirable to have the whole process hinge on the success of a single, central agent. The alternative is to execute the routing process in an asynchronous, decentralized, self-organizing manner. In this case each vertex of the graph is assumed to be a router with limited sensing, processing and decision making capabilities where the immediate domains of awareness and action of a router are limited to a subset of the network with the remaining part being transparent to the router concerned. In other words, the router should sense locally, reason locally, and act locally yet produce global results (figure-9).

In a centralized mode, the routers keep exchanging states till convergence is achieved. The potential is then communicated to a central agent which in a single shot lays a path to the target (figure-10). In a decentralized mode, communication of states between routers need not necessarily be sustained till a steady state is reached. Instead, during communication among the routers, whenever possible, the router with the packet attempt to pass it to a neighboring router using a simple, local, potential-based procedure that can be easily implemented on-board a router. As can be seen, under ideal situation, in a discrete HPF paradigm, the decentralized mode reduces to the centralized one.

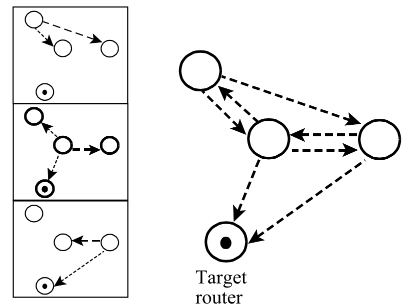


Figure-9: decentralized routing.

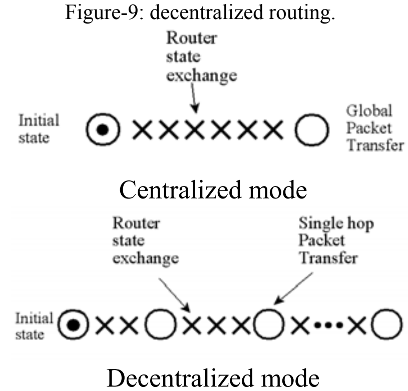


Figure-10: centralized and decentralized mode in a discrete HPF paradigm.

The following is one of the decentralized procedures that may be derived from this paradigm:

0- fix the potential at the target vertex to zero,
1- each router establishes connectivity with selected neighboring routers and assigns appropriate costs,
2- fix the potential at the router that currently hold the data packet to 1,
3- excluding the routers with the packet and the target router, each router should update its potential using equation (18),
4- forward the packet from the current router to the associated router with highest positive flow,
5- if the router is not the target router go to 1,
6- target router is reached.

The procedure is simulated for the graph in figure-2. The potential field was initially set using a random number generator that is uniformly distributed between (0,1). The output of the process is the path: 1 → 6 → 2 → 3 → 5 having the cost 8.

To test the robustness of the procedure the example is repeated while inducing, at each hop, a malfunction in a randomly selected router (excluding the target router and the one currently holding the packet). In the following the vertex number as a function of the hop number is shown for one of the trials (figure-11). As can be seen the packet finally converges to the target vertex. Convergence was observed for all the trials that were carried out.

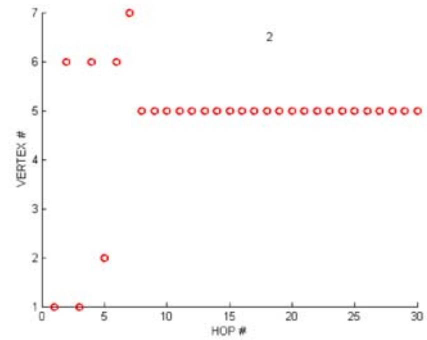


Figure-11: vertex number vs hop number under random router malfunction.

The number of hops needed for the packet to reach the target vertex as a function of the trial number is shown in figure-12 and the corresponding histogram is shown in figure-13.

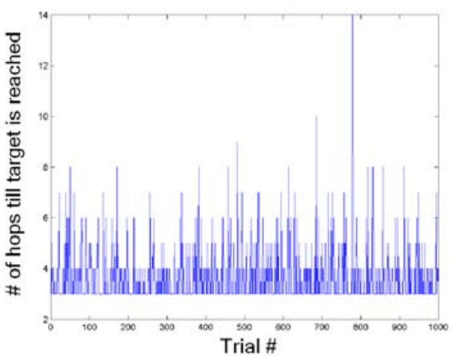


Figure-12: convergence hop umber versus trial number

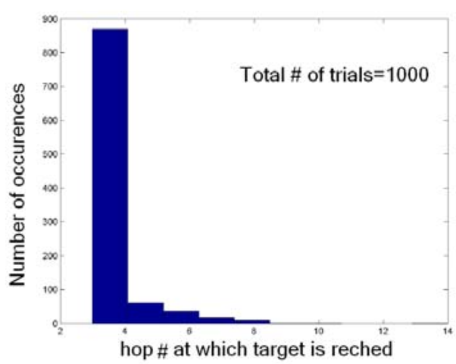


Figure-13: the histogram of figure-12.

VI. Conclusions

In this paper the capability of the harmonic potential field approach to operate in a discrete, sample-based mode is demonstrated. It is shown that a modified, provably-correct, optimal, discrete planning action can be derived using the edge flows obtained from a discrete potential field that is made to satisfy the flow balance conditions represented by Kirchhoff current law while marking the start vertex as a flow source and the target vertex as a drain.

References

[1] A. Masoud, S. Masoud, "Evolutionary Action Maps for Navigating a Robot in an Unknown, Multidimensional, Stationary Environment, Part II: Implementation and Results", the 1997 IEEE International Conference on Robotics and Automation, April 21-27, Albuquerque, New Mexico, USA, pp. 2090-2096.

[2] D. Keymeulen, J. Decuyper, "The Fluid Dynamics Applied to Mobile Robot Motion: the Stream Field Method", The 1994 IEEE International Conference on Robotics and Automation, May 8-13, 1994 San Diego, California, pp. 378-385.

[3] S. Axler, P. Bourdon, W. Ramey, "Harmonic Function Theory", Springer, 1992.

[4] L. Evans, "Partial Differential Equations", American Mathematical Society, 1998.

[5] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential fields. IEEE Trans. Robot. & Autom., 8(5):501-518, October 1992.

[6] S. Masoud, A. Masoud, " Motion Planning in the Presence of Directional and Obstacle Avoidance Constraints Using Nonlinear Anisotropic, Harmonic Potential Fields: A Physical Metaphor", IEEE Transactions on Systems, Man, & Cybernetics, Part A: systems and humans, Vol 32, No. 6, November 2002, pp. 705-723.

[7] A. Masoud "Solving the Narrow Corridor Problem in Potential Field-Guided Autonomous Robots", Submitted to: the 2005 IEEE International Conference On Robotics and Automation, April 18-22, 2005, Barcelona, Spain, pp. 2920-2925.

[8] R. Bohlin, "Robot path planning", Ph.D. thesis, Chalmers University, Gothenburg, Sweden, 2002.

[9] D. Aarno, D. Kragic, H. Christensen, "Artificial Potential Biased Probabilistic Roadmap Method", Proceedings of the 2004 IEEE international conference on robotics and automation, New Orleans, April 2004, pp. 461-466.

[10] M. Kazemi, M. Mehrandezh, K. Gupta, "An Incremental Harmonic Function-based Probabilistic Roadmap Approach to Robot Path Planning", Proceedings of the 2005 IEEE international conference on robotics and automation, Barcelona-Spain, April 2005, pp. 2148-2153.

[11] B. Bollabas, "Graph Theory: An Introductory Course", Springer-Verlag, New York Heidelberg Berlin, 1979.

[12] S. Seshu, M. Reed, "Linear Graphs and Electrical Networks", Addison-Wesley Publishing Company INC., Reading, Massachusetts, USA, London England, 1961.

[13] C. Berge, A. Ghouila-Houri, "Programming, Games and Transportation Networks", London, New York, 1962.

[14] L. Bobrow, "Elementary Linear Circuit Analysis", second edition, OXFORD university press, 1981.

[15] J. Milnor, "Morse Theory", Princeton University Press, 1963.

[16] G. Strang, "Linear Algebra and its Applications", Academic Press, 1988.

[17] A. Masoud, "An Informationally-Open, Organizationally-Closed Control Structure for Navigating a Robot in an Unknown, Stationary Environment" 2003 IEEE International Symposium on Intelligent Control, October 5-8, 2003, Houston, Texas, USA, pp. 614-619.

[18] S. Shapiro, ed., "Encyclopedia of Artificial Intelligence", John Wiley & Sons, New York, 1992.

[19] Ahmad A. Masoud, "A Hybrid, PDE-ODE Control Strategy for Intercepting an Intelligent, Well-Informed Target in a Stationary, Cluttered Environment", accepted for publication at: Applied Mathematical Sciences, HIKARI Ltd, Vol. 1, 2007, no. 48, 2345-2371.

[20] Ahmad A. Masoud, "A Discrete Harmonic Potential Field for Optimum Point-to-point Routing on a Weighted Graph", the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems", October 9-13 2006, Beijing-China, pp. 1779-1784.