The Journal of Engineering Research Vol. 11, No. 2 (2014) 79-88

On Directed Edge-Disjoint Spanning Trees in Product Networks, An Algorithmic Approach

A.R. Touzene* and K. Day

Department of Computer Science, College of Science, Sultan Qaboos University, P.O. Box 36, Postal Code 123, Al-Khodh, Muscat, Sultanate of Oman.

Received 11 December 2013; accepted 15 September 2014

Abstract: In (Ku *et al.* 2003), the authors have proposed a construction of edge-disjoint spanning trees EDSTs in undirected product networks. Their construction method focuses more on showing the existence of a maximum number (n_1+n_2-1) of EDSTs in product network of two graphs, where factor graphs have respectively n_1 and n_2 EDSTs. In this paper, we propose a new systematic and algorithmic approach to construct (n_1+n_2) directed routed EDST in the product networks. The direction of an edge is added to support bidirectional links in interconnection networks. Our EDSTs can be used straight-forward to develop efficient collective communication algorithms for both models store-and-forward and wormhole.

Keywords: Product networks, Directed edge-disjoint spanning trees, Interconnection networks.

نهج خوارزمي : الهيكل الممتد للحد المنفصل الموجه في إنتاج الشبكات

عبدالرزاق توزان* و خالد داي

الملخص: اقترح المؤلفون بناء للهيكل الممتد للحدود المنفصلة EDSTs لإنتاج شبكات غير موجهة. طريقة البناء لديهم تركز أكثر على إظهار وجود الرقم الأقصى (n_1+n_2-1) لأنظمة EDST في إنتاج الشبكات لرسمين بيانيين حيث معامل الرسومات على الترتيب *n1* و *n2*EDSTs. في هذه المقالة نقدم مقترح لمنهج منظم ونظام حسابي جديد لبناء (n_1+n_2) مسار موجهه EDSTs في منتج الشبكات. تم إضافة اتجاه الحد لدعم الروابط الثنائية لشبكات الربط. نظام EDSTs الخاص بنايمكن استخدامه مباشرة لتطوير خوارزميات الاتصالات الجماعية الفعالة لكل من نموذجي التخزين إلى الأمام و الثقب.

مفاتيح الكلمات: منتج الشبكات، الهيكل الممتد للحد المنفصل، الشبكات المتداخلة

*Corresponding author's e-mail: touzene@squ.edu.om

1. Introduction

There has been increasing interest over the last two decades in product networks (Day, and Al-Ayyoub 1997; Ku *et al.* 2003; X and Yang 2007; Imrich *et al.* 2008; Klavar and Špacapan 2008; Jänicke *et al.* 2010; Hammack *et al.* 2011; Chen *et al.* 2011; Ma *et al.* 2011; Cheng *et al.* 2013; Erveš and Žerovnik 2013; Govorčin and Škrekovski 2014). The Cartesian product is a well-known graph operation. When applied to interconnection networks, the Cartesian product operation combines factor networks into a product network. Graph product is an important method to construct bigger graphs, and plays a key role in the design and analysis of networks. A number of spanning trees of a graph are edge-disjoint if no two trees contain the same edge. Edge-Disjoint spanning trees (EDSTs) have many practical applications including enhancing interconnection network fault-tolerance and developing efficient collective communication algorithms in distributed memory parallel computers (Fragopoulo and Akl 1996; Johnsson and Ho 1989; Touzene 2003). In (Ku *et al.* 2003), the authorshave studied construction of maximum edge-disjoint spanning trees(n_1+n_2-1) EDSTs in undirected product network of two graphs, where factor graphs have respectively n_1 and n_2 EDSTs. The presented construction is more about showing the existence of a maximum number of spanning trees. They did not provide a straight-forward algorithmic way for their construction. In this paper, we propose a new systematic and algorithmic approach to construct (n_1+n_2) directed rooted edge-disjoint spanning tree in product networks. We assume that the factor graphs are connected graphs and have respectively n_1 and n_2 EDSTs. Directed rooted edge-disjoint spanning trees have been discussed for different graphs such as the *n*-dimensional hypercube (Johnsson and Ho 1989), k-ary-*n*-cube (Touzene 2003), star graphs (Fragopoulo and Akl 1996), etc. We assume directed edges: if *a* and *b* are two nodes in the graph, the edge *(a, b)* is different from the edge *(b, a)*. Directed edges support bidirectional links in interconnection networks. The advantage of our method is the direct use of our trees to develop collective communication procedures in product interconnection networks.

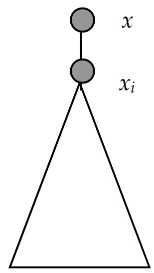
The remainder of this paper is organized as follows: In Section 2, notations and preliminaries are presented. In Section 3, the construction of edge-disjoint spanning trees in product networks is proposed. In Section 4, we conclude this paper.

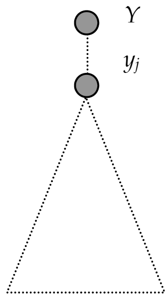
2. Notations and Preliminaries

The Cartesian product $G = G_1 \times G_2$ of two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the undirected graph $G = (V, E)$, where V and E are given by: $V = \{ \langle x_1, x_2 \rangle \mid x_1 \in V_1 \text{ and } x_2 \in V_2\}$, and for any $u = \langle x_1, x_2 \rangle$ and $v = \langle y_1, y_2 \rangle$ in V, (u, v) is an edge in E if, and only if, either (x_1, y_1) is an edge in E_1 and $x_2 = y_2$, or (x_2, y_2) is an edge in E_2 and $x_1 = y_1$. The edge (u, v) is called a G_1-edge if (x_1, y_1) is an edge in G_1, and it is called a G_2-edge if (x_2, y_2) is an edge in E_2. x_1 is called the G_1-component of u and x_2 is called the G_2-component. In all what follows we consider directed edges in the sense that the edge (u, v) is different from the edge (v, u).

3. Construction of EDSTs in a Product Network

Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_1, E_1)$ having the following properties: the graph G_1 contains n_1 EDST all rooted at x denoted: $X_1(x)$, $X_2(x)$, …, $X_{n1}(x)$. Each $X_i(x)$ tree is assumed to be formed of an edge (x, x_i), where x_i is the i^{th} neighbor of x, and a sub-tree denoted $X_i(x)/x$ rooted at x_i that spans all the G_1 nodes other than x (Fig. 1.a). The graph G_2 contains n_2 EDST all rooted at y denoted: $Y_1(y)$, $Y_2(y)$, … , $Y_{n2}(y)$. Each $Y_j(y)$ tree is assumed to be formed of an edge (y, y_j), where y_j is the j^{th} neighbor of y, and a sub-tree denoted $Y_j(y)/y$ rooted at y_j that spans all the G_2 nodes other than y (figure 1.b). In Fig. 1 (a, b) straight lines correspond to G_1-edges and dashed lines correspond to G_2-edges.





X_i (x/x)y$_i$***(y/y)***

Figure 1.a. *i*th EDSTX_i (x) rooted at x in G_1 and its X_i (x) sub-tree.

Figure 1.b. *i*th EDST Y_i (y) at y in G_2 and its Y_j (y)/y sub-tree.

In what follows, we fix a specific node <x_0, y_0> in G as a desired root for the EDST to be constructed. We denote by <x_i, y_0>, i = 1,…, n_1, the n_1 neighbors of <x_0, y_0>in G reached from <x_0, y_0> via G_1-edges, and by <x_0,y_j>, j = 1, …, n_2, the n_2 neighbors of<x_0, y_0> reached from <x_0, y_0>via G_2-edges. For a given node x in G_1 and a given tree Y in G_2, we denote by <x, Y> the tree in G_1×G_2 obtained by fixing the G_1-component to x and following the edges of tree Y in G_2. Similarly, <X, y> denotes the tree in G_1×G_2 obtained by following the edges of a tree X in G_1 while the G_2-component is fixed to node y.

3.1 The ST_1 and ST_2 EDST for G

We present a construction algorithm of n_1+n_2-2 EDST (without using non-tree edges (Ku *et al.* 2003) for the product graph G: n_1-1 EDST for G denoted ST$_1$(i), i = 2.. n_1 and n_2-1EDST for G denoted ST$_2$(j), j = 2.. n_2.

3.2 *Construction of ST$_1$(i), for any i 2 ≤ i ≤ n_1*

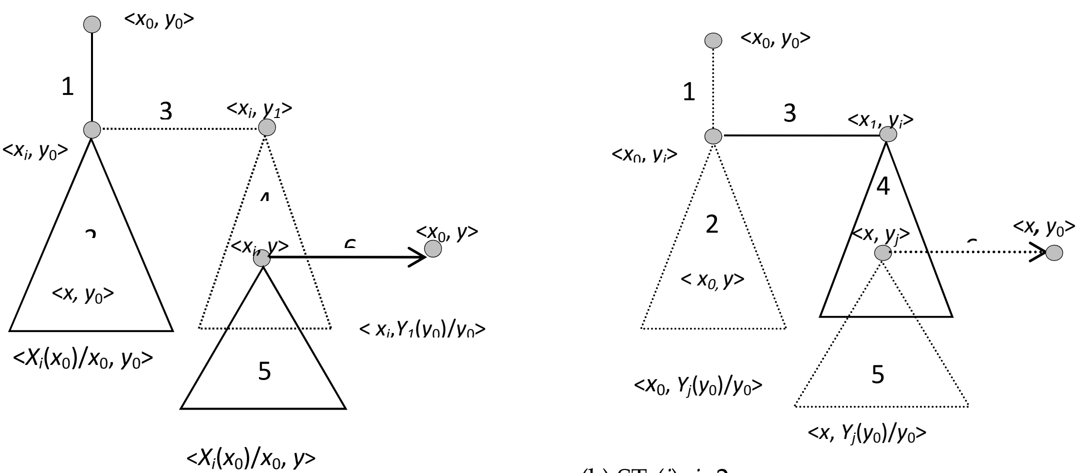
1. Connect <x_0, y_0>to its neighbor <x_i, y_0> (see edge labeled 1 in Fig. 2(a)).
2. Attach to <x_i, y_0> the sub-tree <X_i(x_0)/x_0, y_0>(see sub-tree labeled 2 in Fig. 2(a)).
3. Connect <x_i, y_0> to its neighbor <x_i, y_1> (see edge labeled 3 in Fig. 2(a)).
4. To <x_i, y_1> attach the sub-tree <x_i, Y_1(y_0)/y_0>(see sub-tree labeled 4 in Fig. 2(a)).
5. To each node <x_i, y> in the sub-tree <x_i, Y_1(y_0)/y_0> (including its root <x_i, y_1>) attach the tree <X_i(x_0)/x_0, y>(see sub-tree labeled 5 in Fig. 2(a)).
6. Connect each node <x_i, y> in the sub-tree<x_i, Y_1(y_0)/y_0> (including its root <x_i, y_1>) to its neighbor <x_0, y_1>(see edge labeled 6 in Fig. 2(a)).

3.3 Construction of the tree ST$_2$(j), j=2, .. n_2

1. Connect <x_0, y_0>to its neighbor <x_0, y_j> (see edge labeled 1 in Fig. 2(b)).
2. Attach to <x_0, y_j> the sub-tree <x_0,Y_j(y_0)/y_0>(see sub-tree labeled 2 in Fig. 2(b)).
3. Connect <x_0, y_j> to its neighbor <x_1, y_j> (see edge labeled 3 in Fig. 2(b)).
4. To <x_1, y_j> attach the sub-tree <X_1(x_0)/x_0, y_j> (see labeled 4 in Fig. 2(b)).
5. To each node <x, y_j >in the sub-tree <X_1(x_0)/x_0, y_j > (including its root <x_1, y_j>) attach the tree <x, Y_j(y_0)/y_0> (see sub-tree labeled 5 in Fig. 2(b)).
6. Connect each node <x, y_j >in the sub-tree<X_1(x_0)/x_0, y_j > (including its root <x_1, y_j>) to its neighbor <x_1, y_0>(see edge labeled 6 in figure 2(b)).In figure 2(a, b), straight lines are G_1-edges and dashed lines are to G_2-edges.

Theorem 1: The set {ST$_1$(i), 2 ≤ i ≤ n_2}∪{ST$_2$(j), 2 ≤ j ≤ n_2} is a family of (n_1+n_2-2) edge-disjoint panning trees in G = G_1×G_2.

Proof: We show that all the nodes <x, y> of the product graph are reached in the (n_1+n_2-2) edge-disjoint spanning tree using different edges.



(a) $ST_1(i)$, i=2.. n_1 (b) $ST_2(j)$, j=2.. n_2

Figure 2. Construction of spanning trees ST_1 (i) and ST_2 (j).

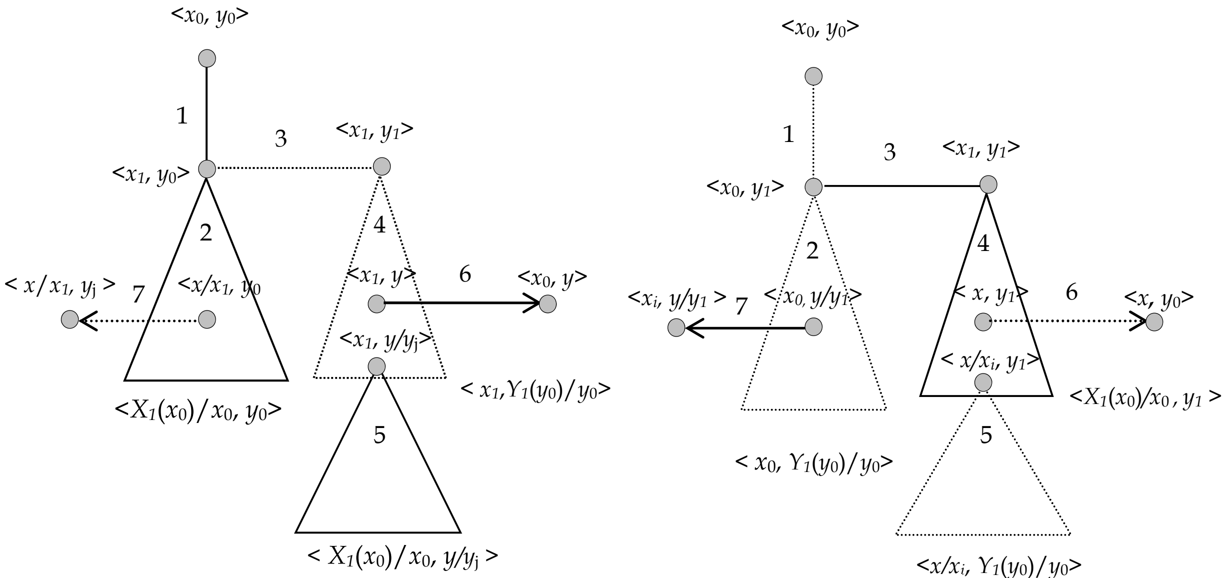
- Case 1: nodes $<x_0, y>$ are reached by different G_1-edges ($<x_i, y>$, $<x_0, y>$), i = 2, ..., n_1, in the different trees $ST_1(i)$ (edges labeled 6 in figure 2(a)). In trees $ST_2(j)$, j = 2, ..., n_2, these nodes are covered by G_2-edges of the sub-trees $<x_0, Y_j(y_0)/y_0>$ (edges labeled 2 in Fig. 2(b)).
- Case 2: nodes $<x, y_0>$, similar proof as in case 1 (symmetrical).
- Case 3: nodes $<x_i, y>$, i = 2, ..., n_1 are covered in four different ways:
  1. In sub-trees $<x_i, Y_1(y_0)/y_0>$, i = 2, ..., n_1 of trees $ST_1(i)$ using Y_1 tree edges (labeled 4 Fig. 2(a)).
  2. In sub-tree $<x, Y_j(y_0)/y_0>$, j = 2, ..., n_2 of the trees $ST_2(j)$. These nodes are covered using Y_j trees edges (j>1), (labeled 5 in Fig. 2(b)).
  3. In sub-trees $<X_i(x_0)/x_0,y>$, i = 2, ..., n_1 of trees $ST_1(j)$ using X_i tree edges (labeled 5 in Fig. 2(a)) .
  4. In sub-tree $<X_1(x_0)/x_0,y_j>$ j = 2, ..., n_2 of the trees $ST_2(j)$ using X_1 tree edges (labeled 4 in Fig. 2(b)).
- Case 4: nodes $<x, y_j>$, similar proof as in case 3 (symmetrical).
- Case 5: nodes $<x, y>$, $x \neq x_i$, $y \neq y_j$ are covered using different G_1-edges in sub-trees $<X_i(x_0)/x_0,y>$, i = 2, ..., n_1 of trees $ST_1(i)$ (sub-tree labeled 5 in Fig. 2(a)). These nodes are covered using G_2-edges in the sub-trees $<x, Y_j(y_0)/y_0>$, j = 2, ..., n_2 in the trees $ST_2(j)$ (labeled 5 in Fig. 2(b)).

3.4 The Special T_1 and T_2 EDSTs for G

We present a construction algorithm for the directed EDSTs in the product graph G denoted T_1 and T_2.

3.5 Construction of T_1

1. Connect $<x_0, y_0>$ to its neighbor $<x_1, y_0>$ (see edge labeled 1 in Fig. 3(a)).
2. Attach to $<x_1, y_0>$ the sub-tree $<X_1(x_0)/x_0, y_0>$ (see sub-tree labeled 2 in Fig. 3(a)).
3. Connect $<x_1, y_0>$ to its neighbor $<x_1, y_1>$ (see edge labeled 3 in Fig. 3(a)).
4. To $<x_1, y_1>$ attach the sub-tree $<x_1, Y_1(y_0)/y_0>$ (see sub-tree labeled 4 in Fig. 3(a)).
5. To each node $<x_1, y/y_j>$, $j=1,\ldots,n_2$ in the sub-tree $<x_1, Y_1(y_0)/y_0>$ (including its root $<x_1, y_1>$) attach the tree $<X_1(x_0)/x_0, y>$ (see sub-tree labeled 5 in Fig. 3(a)).
6. Connect each node $<x_1, y>$ in the sub-tree $<x_1, Y_1(y_0)/y_0>$ (including its root $<x_1, y_1>$) to its neighbor $<x_0, y_1>$ (see edge labeled 6 in Fig. 3(a)).



(a) T_1 **(b) T_2**

Figure 3. Construction of spanning trees T_1 and T_2**.**

7. Connect each node $\langle x_1, y\rangle$ in the sub-tree$\langle x_1, Y_1(y_0)/y_0\rangle$ (including its root $\langle x_1, y_1\rangle$) to its neighbor $\langle x_0, y_1\rangle$(see edge labeled 6 in Fig. 3(a)).
8. Connect each node $\langle x/x_1, y_0\rangle$ in the sub-tree $\langle X_1(x_0)/x_0, y_0\rangle$ to the node $\langle x/x_1, y_j\rangle$(see label 7 in Fig. 3(a)).

3.6 Construction of the Tree T_2

1. Connect $\langle x_0, y_0\rangle$to its neighbor $\langle x_0, y_1\rangle$ (see edge labeled 1 in Fig. 3(b)).
2. Attach to $\langle x_0, y_1\rangle$ the sub-tree $\langle x_0, Y_1(y_0)/y_0\rangle$(see sub-tree labeled 2 in Fig. 3(b)).
3. Connect $\langle x_0, y_1\rangle$ to its neighbor $\langle x_1, y_1\rangle$ (see edge labeled 3 in Fig. 3(b)).
4. To $\langle x_1, y_1\rangle$ attach the sub-tree $\langle X_1(x_0)/x_0, y_1\rangle$(see labeled 4 in Fig. 3(b)).
5. To each node $\langle x/x_i, y_1\rangle$, $i=1,\ldots,n_1$in the sub-tree $\langle X_1(x_0)/x_0, y_1\rangle$ (including its root $\langle x_1, y_1\rangle$) attach the tree $\langle x, Y_1(y_0)/y_0\rangle$ (see sub-tree labeled 5 in Fig. 3(b)).
6. Connect each node $\langle x, y_1\rangle$in the sub-tree$\langle X_1(x_0)/x_0, y_1\rangle$ (including its root $\langle x_1, y_1\rangle$) to its neighbor $\langle x_1, y_0\rangle$.
7. Connect each node $\langle x_0, y/y_1\rangle$ in the sub-tree $\langle x_0, Y_1(y_0)/y_0\rangle$to the node $\langle x_i, y/y_1\rangle$(see label 7 in Fig. 3(b)).

Note that in T_1 the edges ($\langle x, y_0\rangle$, $\langle x, y_j\rangle$) are used but in $T_2(j)$, $2 \le j \le n_2$, the opposite direction edges ($\langle x, y_j\rangle$) , $\langle x, y_0\rangle$) are use. Similarly, in T_2 the edges ($\langle x_0, y\rangle$, $\langle x_i, y\rangle$) are used but in $T_1(i)$, $2 \le i \le n_1$, the opposite direction edges ($\langle x_i, y\rangle$, $\langle x_0, y\rangle$) are used. It is easy to see that using a similar proof as in Theorem 1, the trees T_1 , T_2 , $ST_1(i)$, $2 \le i \le n_2$ and $T_2(j)$, $2 \le j \le n_2$ is a family of (n_1+n_2) directed rooted edge-disjoint spanning trees in $G = G_1 \times G_2$.

To illustrate our construction algorithm, we give a complete example of product of two interconnection networks the 3-cube (3 directed rooted EDTS's (Johnsson and Ho 1989)) and a ring with three nodes (a, b and c) (2 directed rooted EDST's). Dark circles represents the root node of the trees and the numbers on the edges

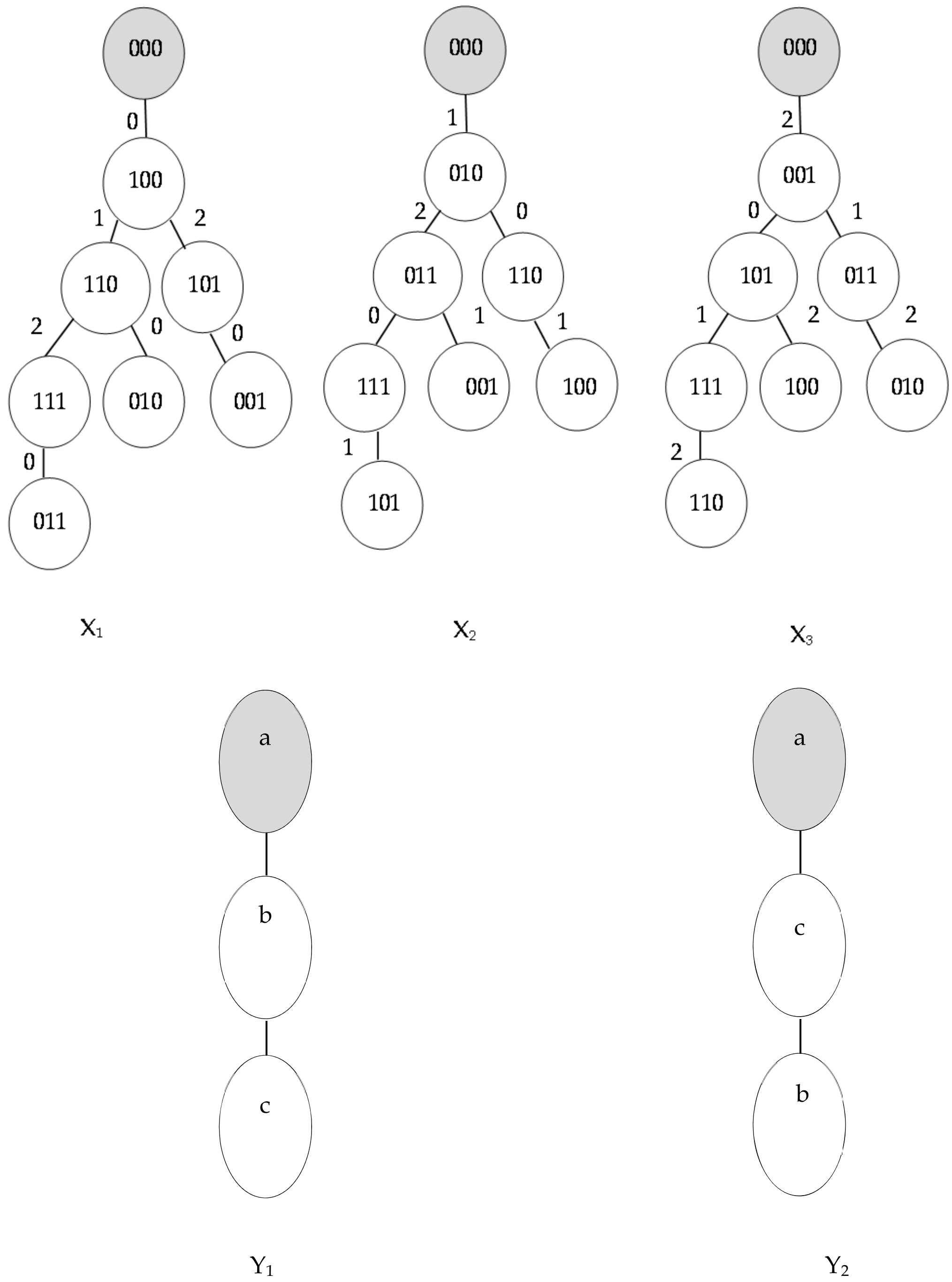


Figure 4. Three EDSTs of the 3-cube and two EDSTs of the ring (3 nodes).

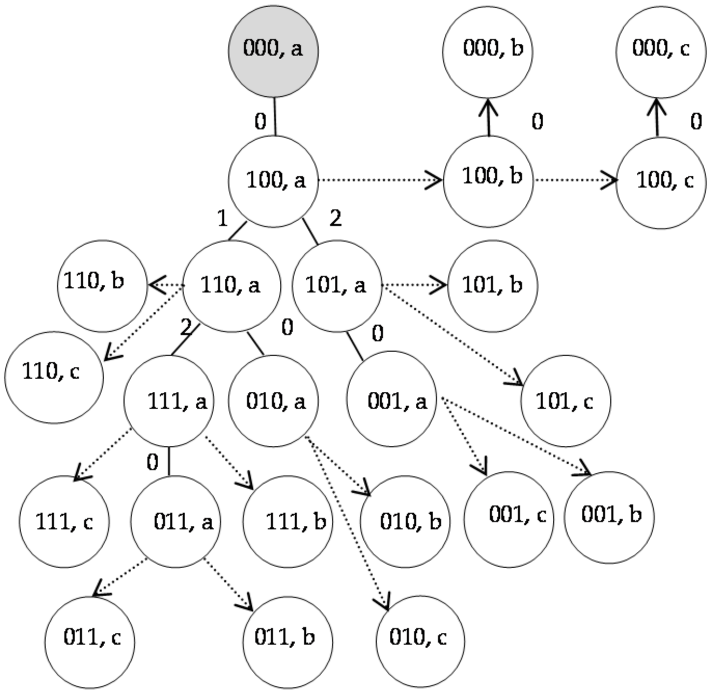


Figure 5 (a).Spanning Tree $ST_1(1)$.

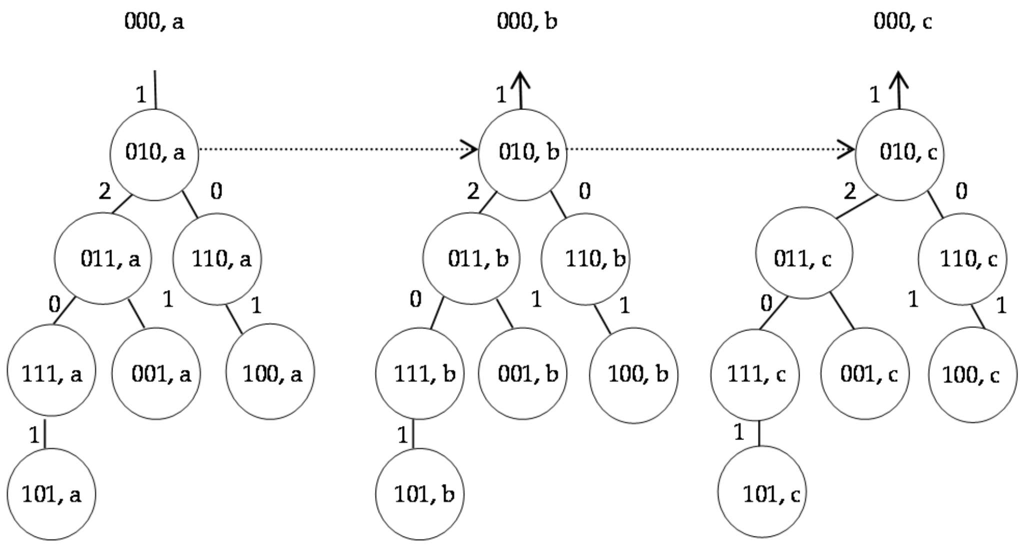


Figure 5 (b). Spanning Tree $ST_1(2)$.

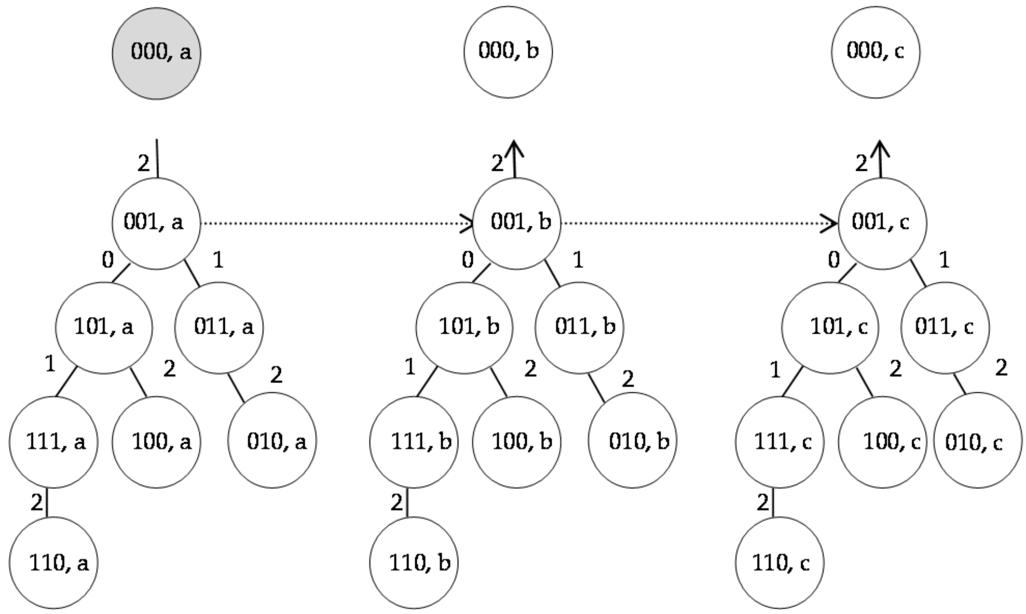


Figure 5 (c). Spanning Tree $ST_1(3)$.

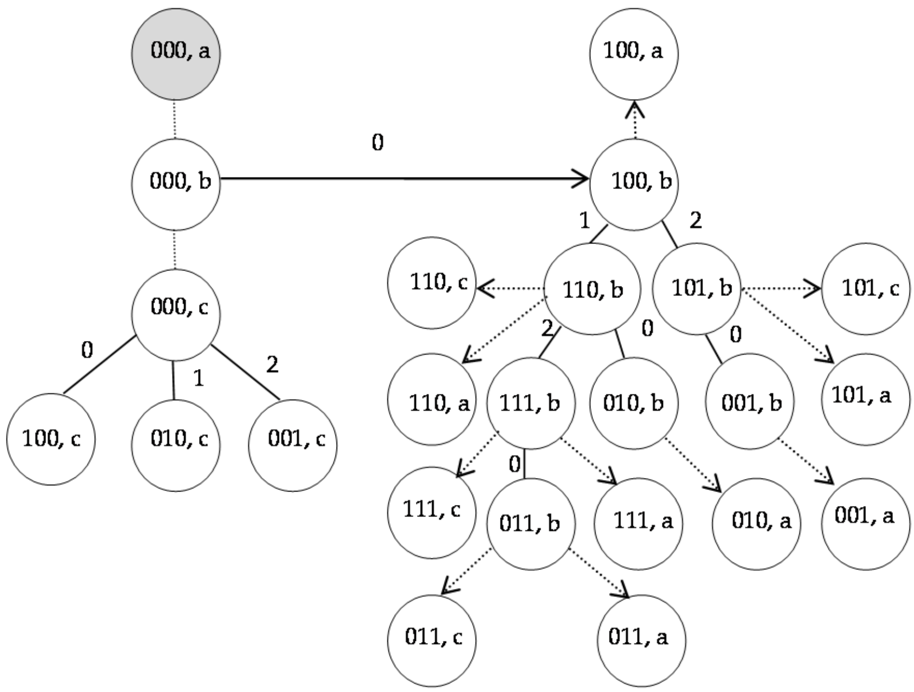


Figure 5 (d). Spanning Tree $ST_2(1)$.

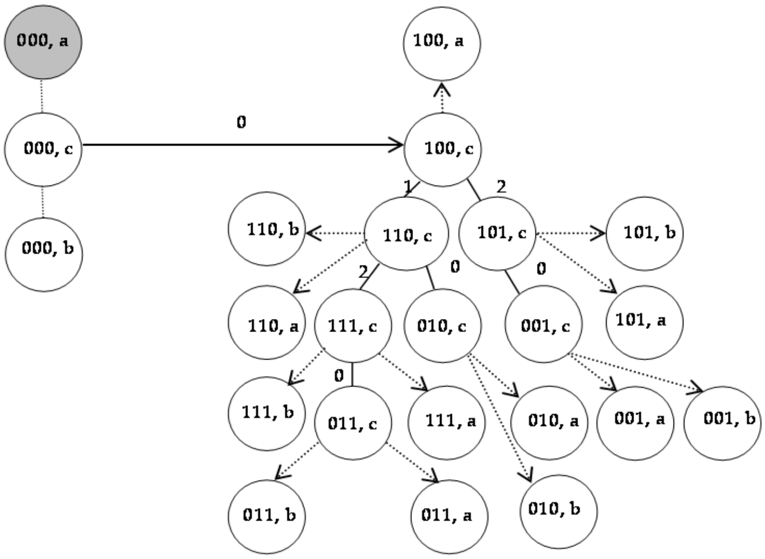


Figure 5 (e). Spanning Tree $ST_2(2)$.

represent the dimension number relative to the 3-cube, see Figs. 4 and 5. The trees are directed from the root nodes to leave nodes.

4. Conclusions

In this paper, we presented a new systematic and algorithmic approach to construct n_1+n_2 (without using non-tree edges) directed rooted edges-disjoint spanning trees for product networks. The previous work on undirected EDSTs of the product networks (Ku *et al.* 2003) focuses more on the existence of n_1+n_2-1 but did not provide an explicit algorithmic way for their construction. Our n_1+n_2 EDSTs can be used straight-forward to develop efficient collective communication algorithms for both models store-and-forward and wormhole using bidirectional links.

References

Chen M, GuoX, Zhai S (2011), The optimal strong radius and optimal strong diameter of the Cartesian product graphs. Applied Mathematics Letters 24(5):657-660.

Cheng CW, Lee CW, Hsieh SY (2013), Conditional edge-fault hamiltonicity of cartesian product graphs. IEEE Transactions on Parallel and Distributed Systems 24 (10):1951-1960.

Day K, Al-Ayyoub AE (1997) , The Cross Product of Interconnection Networks. IEEE Trans. Parallel and Distributed Systems 8(2):109-118.

Erveš R, Žerovnik J (2013), Mixed fault diameter of Cartesian graph bundles. Discrete Applied Mathematics 161 (12):1726-1733.

Fragopoulo P, Akl SG (1996), Edge-disjoint spanning trees on the star network with application to fauttolerance. IEEE Trans. Computers 45(2):174-185.

Govorčin J, Škrekovski R (2014) , On the connectivity of Cartesian product of graphs. ArsMathematicaContemporanea 7(2):293-297.

Hammack R, Imrich W, Klavžar S (2011), Handbook of Product Graphs,*Discrete Mathematics and Its Applications,* Second ed. CRC Press Boca Raton.

Imrich W, Klavžar S, Rall DF (2008) , Topics in Graph Theory: Graphs and their Cartesian Product, AK Peters, Ltd., Wellesley, MA.

Jänicke S, Heine C, Hellmuth M, Stadler PF, Scheuermann G (2010), Visualization of graph products. IEEE Transactions on Visualization and Computer Graphics 16(6):1082-1089.

Johnsson SL, Ho CT (1989) , Optimal broadcasting and personalized communication in hypercubes. IEEE Trans. Computers 38(9):1249-1268.

Klavar S, Špacapan S (2008), On the edge-connectivity of cartesian product graphs. Asian-European Journal of Mathematics 1(1):93-98.

Ku S, Wang B , Hung T (2003), Constructing edge-disjoint spanning trees in product networks. IEEE Trans. Parallel and Distributed Systems 14(3):213-221.

Ma M,Xu JM, Zhu Q (2011),The menger number of the cartesian product of graphs. Applied Mathematics Letters 24(5):627-629.

Touzene A (2004), Optimal all-port collective communication algorithms for the k-ary n-cube interconnection networks. Journal of Systems Architecture 50:221-231.

Xu JM, Yang C(2007), Fault diameter of product graphs. Information Processing Letters 102(6):226-228.