

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-9804

REVISITING HAMILTONIAN  
DECOMPOSITION OF THE HYPERCUBE

K. Okuda and S.W. Song

dezembro de 1998

# Revisiting Hamiltonian Decomposition of the Hypercube

K. Okuda and S. W. Song

University of São Paulo

Institute of Mathematics and Statistics

Department of Computer Science

Rua do Matão, 1010

05508-900, São Paulo, SP, Brazil

Tel: +55 11 818 6101 Fax: +55 11 814 4135

e-mail: [kunio@ime.usp.br](mailto:kunio@ime.usp.br), [song@ime.usp.br](mailto:song@ime.usp.br)

Contact author: S. W. Song

## Abstract

The Hamiltonian decomposition of a hypercube or binary  $n$ -cube is the partitioning of its edge set into Hamiltonian cycles. It is known that there are  $\lfloor n/2 \rfloor$  disjoint Hamiltonian cycles on a binary  $n$ -cube. The proof of this result, however, does not give rise to any simple construction algorithm of such cycles. In a previous work Song presents ideas towards a simple and interesting method to this problem. Two phases are involved. First decompose the binary  $n$ -cube into cycles of length 16,  $C_{16}$ , and then apply a merge operator to join the  $C_{16}$  cycles into larger Hamiltonian cycles. The case of dimension  $n = 6$  (a 64-node hypercube) is illustrated. He conjectures the method can be generalized for any even  $n$ . In this paper, we generalize the first phase of that method for any even  $n$  and prove its correctness. Also we show four possible merge operators for the case of  $n = 8$  (a 256-node hypercube). This result can be viewed as a step toward the general merge operator, thus proving the conjecture.

## 1 Introduction

Many results on the existence of Hamiltonian cycles in graphs are known [2, 3, 4, 5, 6]. In particular the existence of Hamiltonian cycles on the hypercube is important for fault-tolerant distributed computing based on this structure. When an application uses processors joined as a cycle, it is important to know alternative cycles in case of communication failure in one cycle [1]. The mere knowledge of its existence, however, may not be sufficient. It is desirable to have a simple algorithm to construct the alternative cycles.

It is known that there are  $\lfloor n/2 \rfloor$  disjoint Hamiltonian cycles on a hypercube of dimension  $n$  [2]. The proof of this result, however, does not give rise to any simple construction algorithm of such cycles. In [7] Song presents ideas towards a simple and interesting method to this problem. Two phases are involved. (1) Decompose the hypercube into  $C_{16}$  (cycles of length 16) and then (2) apply a merge operator to join the obtained  $C_{16}$  cycles into larger cycles. An illustration for the case of dimension  $n = 6$  (a 64-node hypercube) is given. He conjectures this method can be generalized for any even  $n$ .

In this paper, we generalize the first phase of that method for any even  $n$  and prove its correctness. Also we show a merge operator for the case of  $n = 8$  (a 256-node hypercube).

This result can be viewed as a step toward the general merge operator, thus proving the conjecture.

In Section 2 we discuss Hamiltonian decomposition of the hypercube and present Song's method. In Section 3 we formalize phase 1 for the general case and prove the correctness. In Section 4 we show a merge operator to decompose the hypercube of dimension 8 into Hamiltonian cycles. Section 5 contains the conclusion.

## 2 Previous works

### 2.1 Hamiltonian decomposition

The Hamiltonian decomposition of a graph is the partitioning of its edge set into Hamiltonian cycles. Obviously in order for this partition to exist, the graph must be regular and have even degree. A more general definition of Hamiltonian decomposition is given in [2] to include the case of odd degree. This work considers the Hamiltonian decomposition of the binary  $n$ -cube or hypercube of dimension  $n$ .

For simplicity and without loss of generality, we consider  $n$  to be even. (If  $n$  is odd, the edge set can be partitioned into  $(n-1)/2$  Hamiltonian cycles and a perfect matching [2].) Observe first that the binary  $n$ -cube is equivalent to a 4-ary  $n/2$ -cube, that is the Cartesian product of  $n/2$  cycles of length 4:  $C_4 \times C_4 \times \dots \times C_4$ . We start with the following theorem (see [2] for details and proof).

**Theorem 1** *The binary  $n$ -cube with even  $n$ , or equivalently the product of  $n/2$  cycles,  $C_4 \times C_4 \times \dots \times C_4$ , can be partitioned into  $n/2$  Hamiltonian cycles.*

Song [7] presents ideas and that may give rise to a very simple method to construct the disjoint Hamiltonian cycle of a binary  $n$ -cube. He illustrated the method for the case of  $n = 6$ . It consists of two phases.

1. partition the edge set into cycles of length 16 or  $C_{16}$ .
2. merge the resulting cycles into larger cycles to get the desired Hamiltonian cycles.

This second phase is realized by using a merge operator to be seen later.

### 2.2 Phase 1 – decomposition into $C_{16}$

Phase 1 decomposes the  $n$ -cube into cycles of length 16, or  $C_{16}$ . Foregger [5] gave a solution for the case of  $n = 4$ , i.e.  $C_4 \times C_4$  is decomposed into two  $C_{16}$ .

For the case  $n = 6$ , phase 1 consists of decomposing  $C_4 \times C_4 \times C_4$  into 12  $C_{16}$ . We repeat here how this decomposition is done.

We divide the 12 cycles into three groups:

- 4 cycles of color 0
- 4 cycles of color 1
- 4 cycles of color 2

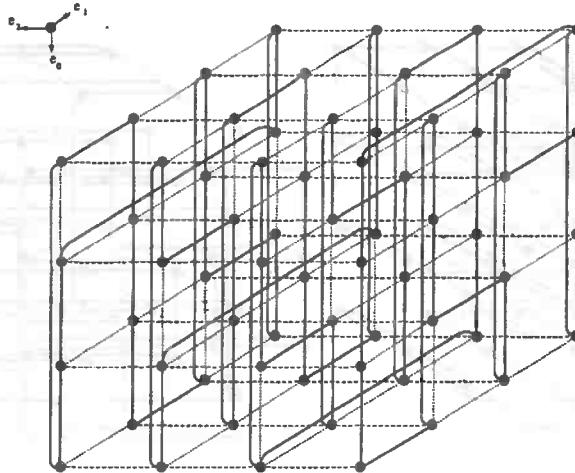


Figure 1: Cycles of color 0

In the next section the cycles  $C_{16}$  of same color will be merged to form a Hamiltonian cycle.

In Figure 1 we illustrate the 4 cycles of color 0. Notice that the cycles are situated on planes parallel to the plane defined by the axes  $e_0$  and  $e_1$ . This observation will be formalized in the next section. In Figure 2 we have the cycles of color 1 and color 2. Notice the symmetries between cycles of this figure and those of Figure 1.

The edge set of the product  $C_4 \times C_4 \times \cdots \times C_4$  ( $n/2$  times) can be partitioned into  $n2^n/32$  disjoint cycles of length 16,  $2^n/16$  cycles of the same color.

### 2.3 Phase 2 – the merge operator

We show how the 12 cycles of length 16 of the previous section can be merged into 3 Hamiltonian cycles. Cycles of the same color will be merged together to form one Hamiltonian cycle.

Consider a vertex and the edges incident with it. An edge permutation operator is an operator that permutes the colors of the edges. We use a set of edge permutation operators to merge cycles of a given color to form a large cycle of the same color.

**Definition 1** *A set of edge permutation operators is a (cycle) merge operator if it transforms a partition of the edge set of  $r$  cycles to a partition of the edge set of  $s$  cycles ( $s < r$ ).*

Consider the partition of the edges of  $C_4 \times C_4 \times C_4$  into  $C_{16}$  as before. Figure 3 shows a merge operator that joins two cycles of each color into a large cycle of the same color. Point A is a reference point for the application of the merge operator. Figure 4 shows the effect of applying the merge operator. For each color the curves (i.e. the additional part

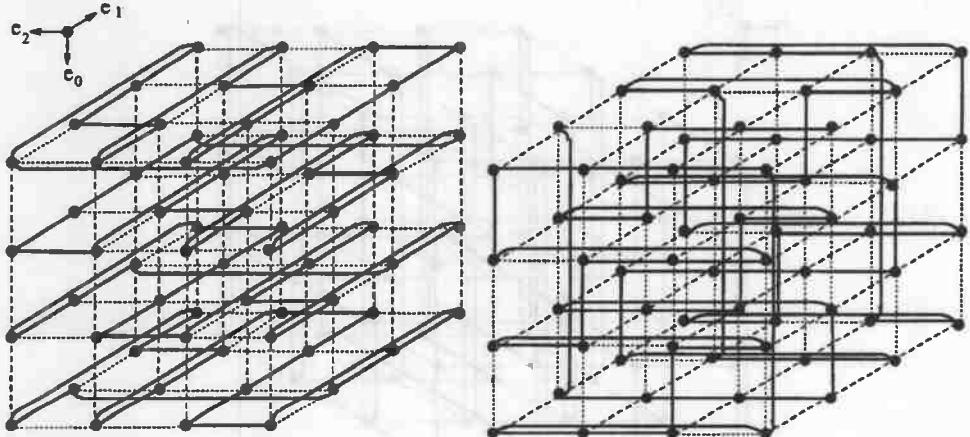


Figure 2: Cycles of color 1 (left) and of color 2 (right)

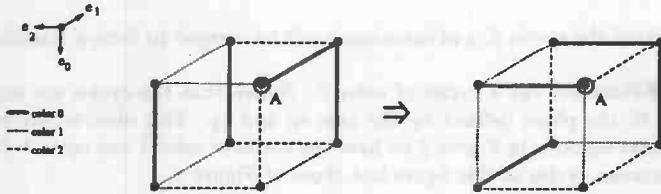


Figure 3: Merge operator

not present in Figure 3) indicate the remaining of the  $C_{16}$  cycle. On the left of Figure 5 we have the positions (indicated by larger circles) of possible reference points **A** to apply the merge operator, according to Figure 3. On the right of the same figure we have the three points chosen in [7] resulting in 3 Hamiltonian cycles of length 64, one for each color 0, 1 and 2.

### 3 Generalization of phase 1

In the following we formalize phase 1, the decomposition of the edge set of a binary  $n$ -cube, for any even  $n$ , into cycles  $C_{16}$ . Let  $n = 2m$  and  $H_m$  a binary  $n$ -cube.

$$H_m = \underbrace{C_4 \times C_4 \times \cdots \times C_4}_{m \text{ times}}$$

Consider a vertex  $x$  of  $H_m$ . Let  $x = (x_0, x_1, \dots, x_{m-1})$ , where  $x_i \in \{0, 1, 2, 3\}$  for  $0 \leq i \leq m-1$ .

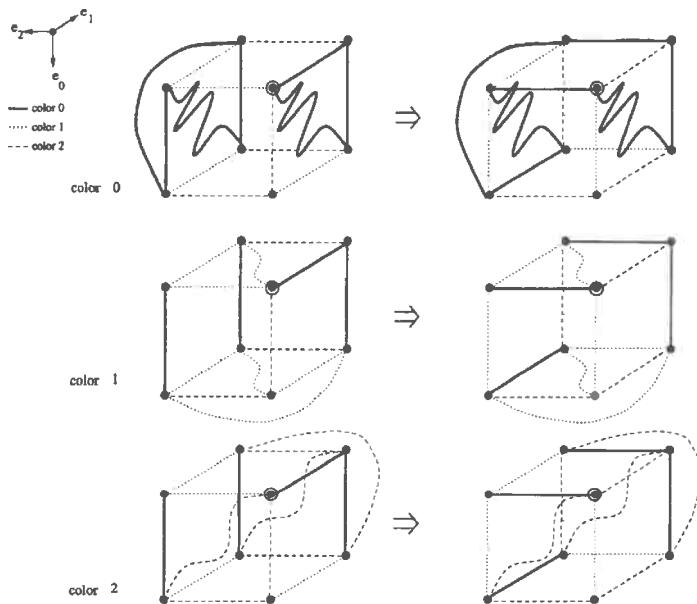


Figure 4: The effect of the merge operator on colors 0, 1 and 2

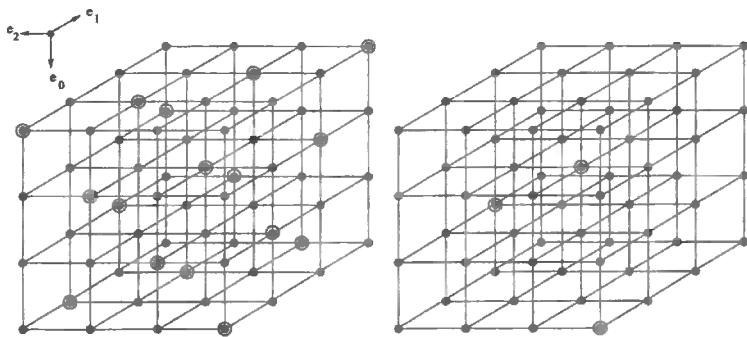


Figure 5: Candidate points to apply the merge operator (left) and the three chosen points (right)

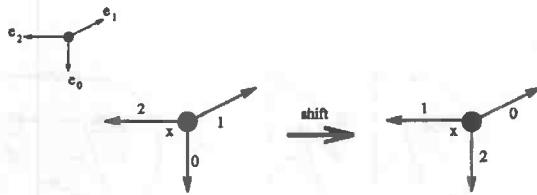


Figure 6: The shift operator for the case  $n = 6$

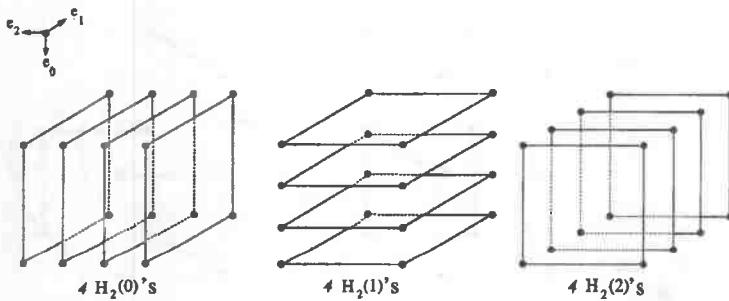


Figure 7: The  $12 H_2$ 's for  $H_3$

Denote by  $x(i)$  the edge joining  $x = (x_0, x_1, \dots, x_i, \dots, x_{m-1})$  and  $x' = (x_0, x_1, \dots, x_i + 1 \pmod 4, \dots, x_{m-1})$ .

**Definition 2** *The initial coloring of the edge set of a binary  $n$ -cube is defined as follows. Color each edge  $x(i)$  with color  $i$ ,  $\forall x \in H_m$  and  $0 \leq i \leq m - 1$ .*

Thus each edge parallel to axis  $e_i$  is colored by color  $i$  with the initial coloring.

**Definition 3** *The shift operator is an operator that, applied to a vertex  $x \in H_m$ , defines the colors of half of the edges incident with  $x$  in the following manner: each  $x(i)$  is colored with color  $(i - 1) \pmod m$ , for  $0 \leq i \leq m - 1$ .*

The above operator is named *shift operator* because, given a vertex  $x$  and an initial coloring of the edges, the application of the shift operator to  $x$  has the effect of *shifting* the colors. See Figure 6 for  $n = 6$ . (Note however that the shift operator does not really shift the edge colors but always gives the same coloring as defined, independent of the initial colors before its application.)

Thus, given an initial coloring of the edges, by applying the shift operator at some points, say  $x$ , of  $H_m$ , edges  $x(i + 1 \pmod m)$  parallel to axis  $e_{i+1 \pmod m}$  will have color

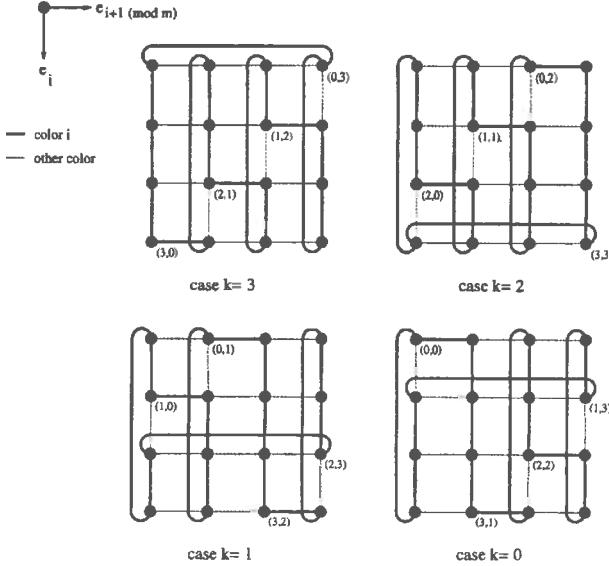


Figure 8: Illustration of Lemma 1

$i$  but all the other edges also with color  $i$  are parallel to  $e_i$ . Furthermore, after applying shift, any path formed by edges of color  $i$  only lies within a  $H_2 = C_4 \times C_4$  parallel to the plane defined by axes  $e_i$  and  $e_{i+1} \pmod{m}$ . This observation takes us to the following definition.

**Definition 4** For each  $0 \leq i \leq m-1$ ,  $H_2(i)$  is a  $H_2$  parallel to the plane defined by axes  $e_i$  and  $e_{i+1} \pmod{m}$ .

For example, for  $n=6$ ,  $H_3 = C_4 \times C_4 \times C_4$  has 12  $H_2(i)$ 's, with 4  $H_2(i)$ 's for each  $i = 0, 1, 2$  (see Figure 7).

A vertex  $x = (x_0, x_1, \dots, x_i, x_{i+1} \pmod{m}, \dots, x_{m-1})$  of  $H_2(i)$  have constant coordinates except  $x_i$  and  $x_{i+1} \pmod{m}$ . It is therefore of the following type:

$$x = (\text{constant}, \dots, \text{constant}, x_i, x_{i+1} \pmod{m}, \text{constant}, \dots, \text{constant})$$

To simplify the notation, we omit the  $(m-2)$  constant coordinates of  $H_2(i)$  and write  $x = (x_i, x_{i+1} \pmod{m})$ .

The following lemma will be used in the proof of Theorem 2.

**Lemma 1** Consider a  $H_2(i)$ , for a fixed  $i$ ,  $0 \leq i \leq m-1$ , a fixed  $k \in \{0, 1, 2, 3\}$ , and an initial coloring. Apply shift in this  $H_2(i)$  to all  $x = (x_i, x_{i+1} \pmod{m})$  such that  $x_i + x_{i+1} \pmod{m} = k \pmod{4}$ . Then the edges of color  $i$  in this  $H_2(i)$  form a  $C_{16}$ .

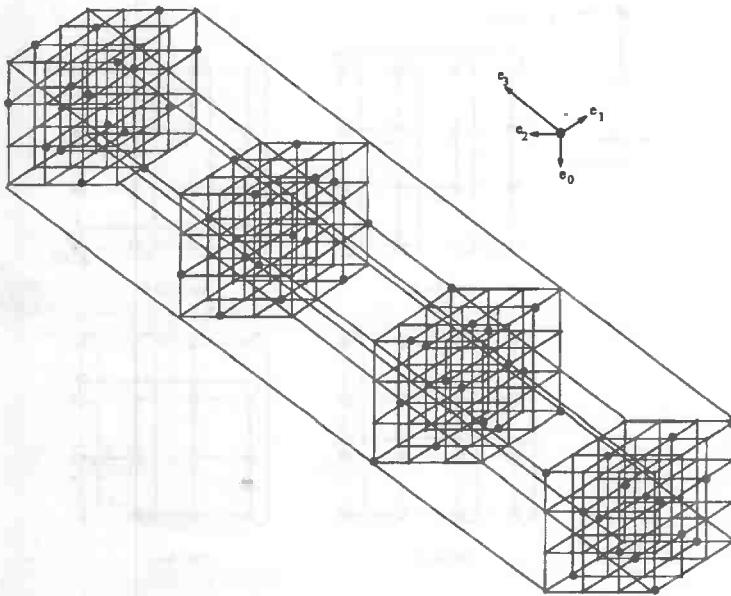


Figure 9: Points to apply the shift operator for  $H_4$

**Proof.** The proof is straightforward. We consider all possible cases for  $k$  ( $k = 0, 1, 2, 3$ ) and obtain Figure 8. The lemma holds.  $\square$

The next theorem generalizes the decomposition in cycles  $C_{16}$  needed in phase 1 of the method of [7].

**Theorem 2** Consider a binary  $n$ -cube  $H_m$  ( $n = 2m$ ), with the initial coloring of its edges, and a fixed  $K \in \{0, 1, 2, 3\}$ . Apply the shift operator to all vertices  $x = (x_0, x_1, \dots, x_{m-1})$  such that  $x_0 + x_1 + \dots + x_{m-1} = K \pmod{4}$ . Then  $H_m$  is decomposed into cycles  $C_{16}$ , i.e., any edge is part of a cycle  $C_{16}$ .

**Proof.** Consider any edge of color  $i$ ,  $0 \leq i \leq m-1$ . It belongs to some  $H_2(i)$ . In this  $H_2(i)$ , consider the vertices at which the shift operator has been applied. As the  $(m-2)$  vertex coordinates of this  $H_2(i)$  are constant, the difference between  $K$  and the sum of these constants will also be constant. Call this difference  $d$ . Then the shift operator was applied at vertices  $x$  such that  $x_i + x_{i+1} \pmod{m} = d \pmod{4}$  in this  $H_2(i)$ . This is exactly the situation described in Lemma 1. Therefore in this  $H_2(i)$  this edge must be part of the cycle  $C_{16}$  of color  $i$ .  $\square$

Note that the decomposition in cycles  $C_{16}$  for  $n = 6$  presented in [7] is a special case of this theorem with  $K = 3$ .

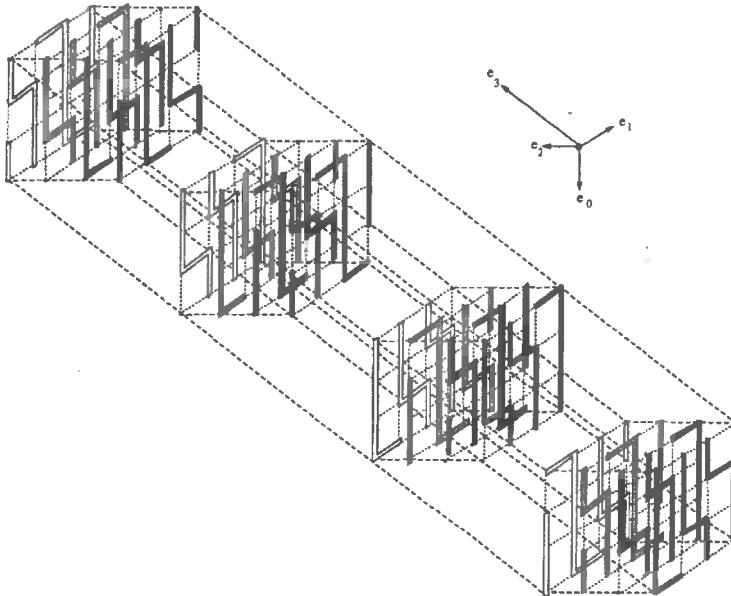


Figure 10: 16 cycles  $C_{16}$  of color 0

## 4 Hamiltonian decomposition for $n = 8$

We now present a Hamiltonian decomposition of the binary  $n$ -cube for  $n = 8$ .

### 4.1 Phase 1: decomposing $H_4$ into 64 cycles $C_{16}$

Apply Theorem 2 for  $H_4$  with  $K = 3$ . (The choice of  $K = 3$  is that applying the theorem for  $H_3$  with this  $K$  would produce the same results and illustrations as in [7] for  $H_3$ .) In Figure 9 the dark points are those to apply the shift operator. For clarity of illustration, some of the edges are omitted in the figure (most of the edges of type  $x(3)$  and all edges of type  $x(i)$ ,  $x = (x_0, x_1, x_2, x_3)$ , with  $x_i = 3$ ).  $H_4$  is decomposed into 64 cycles  $C_{16}$ , with 16  $C_{16}$  of color  $i$  for each  $i = 0, 1, 2, 3$ . In Figure 10 the 16 cycles  $C_{16}$  of color 0 are illustrated. In this figure we omit more edges than we did in the previous figure, for the sake of clarity. Compare this figure with Figure 8 to verify the omitted edges of color 0 that are part of the cycle.

### 4.2 Phase 2: obtaining a merge operator

We found 4 merge operators (see Figure 11 and Figure 12). A small example is given in Figure 13 to illustrate the effect of applying the merge operator. We choose the merge operator 1 and use color 0. On the left of this figure, the large black point is the point where the shift operator has been applied and the bold curves indicate the remaining of

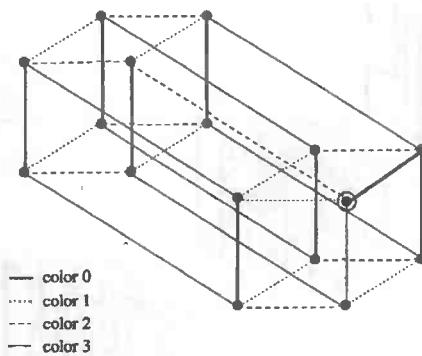


Figure 11: Before applying the merge operator

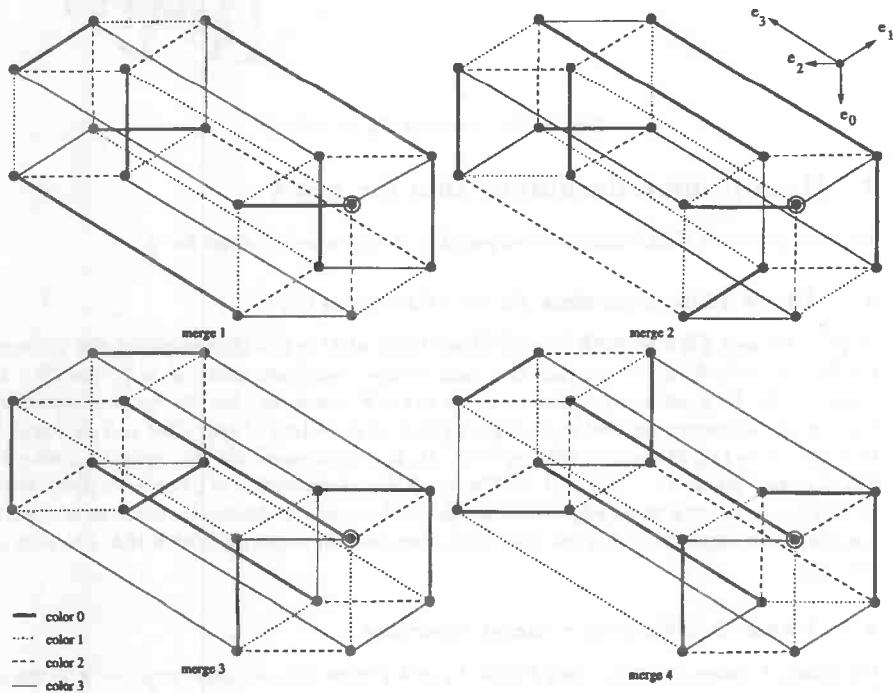


Figure 12: After applying each of the 4 merge operators for  $n = 8$

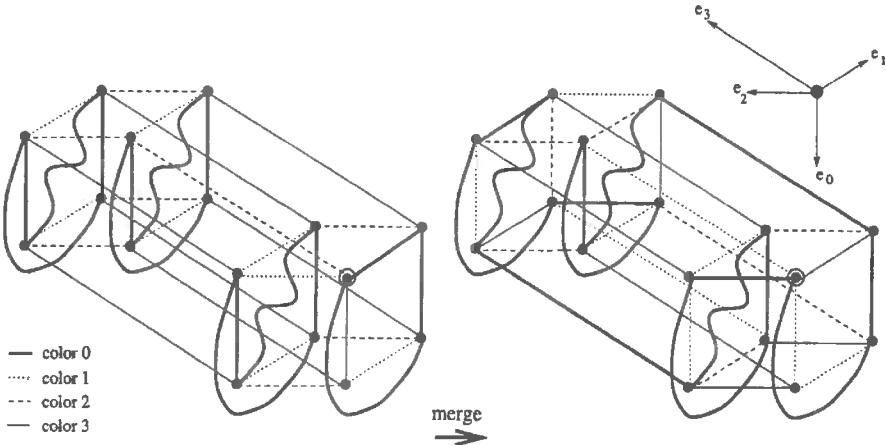


Figure 13: Effect of the merge operator 1 on color 0

the cycle  $C_{16}$  of color 0 in  $H_2(0)$ . We can observe four cycles of color 0. On the right we notice that the merge operator (merge 1) joins the four cycles of color 0 into one sole cycle. The same effect can be observed for colors 1, 2, and 3, that is, four cycles of each color are joined by applying merge 1. The same is valid for the other merge operators (merge 2, merge 3 and merge 4). We apply the merge operator at the following five reference points:  $(3, 0, 0, 0), (1, 0, 2, 0), (0, 1, 1, 1), (3, 2, 0, 2), (1, 2, 2, 2)$  and we join the 16 cycles of each color  $i = 0, 1, 2, 3$  into one cycle only obtaining four cycles  $C_{256}$ .

## 5 Conclusion

We continue the work initiated by Song[7] of finding a simple constructive algorithm to obtain the disjoint Hamiltonian cycles of a binary  $n$ -cube. The previous work consisted of the following. First decompose the binary  $n$ -cube into cycles of length 16,  $C_{16}$ , and then apply a merge operator to join the  $C_{16}$  cycles into larger Hamiltonian cycles. The case of dimension  $n = 6$  (a 64-node hypercube) was illustrated. The conjecture was that the method can be generalized for any even  $n$ .

In this paper we generalize the first phase of that method for any even  $n$  and prove its correctness. Also we show four possible merge operators for the case of  $n = 8$  (a 256-node hypercube). This result can be viewed as a step toward the general merge operator, thus proving the conjecture.

## References

[1] J. Armstrong and F. G. Gray. Fault diagnosis in a Boolean  $n$ -cube array of microprocessors. *IEEE Transactions on Computers* **38** (1981) 587 - 590.

- [2] B. Alspach, J.-C. Bermond, D. Sotteau. Decomposition into cycles I: Hamilton decompositions. In G. H. et al. (editors) *Cycles and rays*. Kluwer Academic, 9 - 18, 1990.
- [3] J. Aubert et B. Schneider. Décomposition de  $K_m + K_n$  en cycles Hamiltoniens. *Discrete Mathematics* 37 (1981) 19 - 27.
- [4] J. Aubert et B. Schneider. Décomposition de la somme Cartesienne d'un cycle et de l'union de deux cycles Hamiltoniens en cycles Hamiltoniens. *Discrete Mathematics* 38 (1982) 7 - 16.
- [5] M. F. Foregger. Hamiltonian decompositions of products of cycles. *Discrete Mathematics* 24 (1978) 251 - 260.
- [6] A. Kotzig. Every Cartesian product of two circuits is decomposable into two Hamiltonian circuits. Centre de Recherche Mathématiques, Montreal (1973).
- [7] S. W. Song. Towards a simple construction method for Hamiltonian decomposition of the hypercube. *Discrete Mathematics and Theoretical Computer Science*, Vol. 21, DIMACS Series, American Mathematical Society, 1995, pp. 297 - 306.

**RELATÓRIOS TÉCNICOS**  
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**  
Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1994 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail([mac@ime.usp.br](mailto:mac@ime.usp.br)).

Thomaz I. Seidman e Carlos Humes Jr.

*SOME KANBAN-CONTROLLED MANUFACTURING SYSTEMS: A FIRST STABILITY ANALYSIS*  
RT-MAC-9501, janeiro de 1995, 19 pp.

C. Humes Jr. and A.F.P.C. Humes

*STABILIZATION IN FMS BY QUASI-PERIODIC POLICIES*  
RT-MAC-9502, março de 1995, 31 pp.

Fabio Kon e Arnaldo Mandel

*SODA: A LEASE-BASED CONSISTENT DISTRIBUTED FILE SYSTEM*  
RT-MAC-9503, março de 1995, 18 pp.

Junior Barrera, Nina Sumiko Tomita, Flávio Soares C. Silva, Routh Terada

*AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY PAC LEARNING*  
RT-MAC-9504, abril de 1995, 16 pp.

Flávio S. Corrêa da Silva e Fabio Kon

*CATEGORIAL GRAMMAR AND HARMONIC ANALYSIS*  
RT-MAC-9505, junho de 1995, 17 pp.

Henrique Mongelli e Routh Terada

*ALGORITMOS PARALELOS PARA SOLUÇÃO DE SISTEMAS LINEARES*  
RT-MAC-9506, junho de 1995, 158 pp.

Kunio Okuda

*PARALELIZAÇÃO DE LAÇOS UNIFORMES POR REDUÇÃO DE DEPENDÊNCIA*  
RT-MAC-9507, julho de 1995, 27 pp.

Valdemar W. Setzer e Lowell Manke

*COMPUTERS IN EDUCATION: WHY, WHEN, HOW*  
RT-MAC-9508, julho de 1995, 21 pp.

Flávio S. Corrêa da Silva

*REASONING WITH LOCAL AND GLOBAL INCONSISTENCIES*  
RT-MAC-9509, julho de 1995, 16 pp.

Julio M. Stern

*MODELOS MATEMÁTICOS PARA FORMAÇÃO DE PORTFÓLIOS*  
RT-MAC-9510, julho de 1995, 43 pp.

Fernando Iazzetta e Fabio Kon

*A DETAILED DESCRIPTION OF MAXANNEALING*  
RT-MAC-9511, agosto de 1995, 22 pp.

Flávio Keidi Miyazawa e Yoshiko Wakabayashi  
*POLYNOMIAL APPROXIMATION ALGORITHMS FOR THE ORTHOGONAL  
Z-ORIENTED 3-D PACKING PROBLEM*  
RT-MAC-9512, agosto de 1995, pp.

Junior Barrera e Guillermo Pablo Salas  
*SET OPERATIONS ON COLLECTIONS OF CLOSED INTERVALS AND THEIR APPLICATIONS TO THE  
AUTOMATIC PROGRAMMING OF MORPHOLOGICAL MACHINES*  
RT-MAC-9513, agosto de 1995, 84 pp.

Marco Dimas Gubitoso e Jörg Cordsen  
*PERFORMANCE CONSIDERATIONS IN VOTE FOR PEACE*  
RT-MAC-9514, novembro de 1995, 18pp.

Carlos Eduardo Ferreira e Yoshiko Wakabayashi  
*ANÁIS DA I OFICINA NACIONAL EM PROBLEMAS COMBINATÓRIOS: TEORIA, ALGORITMOS E  
APLICAÇÕES*  
RT-MAC-9515, novembro de 1995, 45 pp.

Markus Endler and Anil D'Souza  
*SUPPORTING DISTRIBUTED APPLICATION MANAGEMENT IN SAMPA*  
RT-MAC-9516, novembro de 1995, 22 pp.

Junior Barrera, Routh Terada,  
Flávio Corrêa da Silva and Nina Sumiko Tomita  
*AUTOMATIC PROGRAMMING OF MMACH'S FOR OCR\**  
RT-MAC-9517, dezembro de 1995, 14 pp.

Junior Barrera, Guillermo Pablo Salas and Ronaldo Fumio Hashimoto  
*SET OPERATIONS ON CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC  
PROGRAMMING OF MMACH'S*  
RT-MAC-9518, dezembro de 1995, 14 pp.

Daniel V. Carbogim and Flávio S. Corrêa da Silva  
*FACTS, ANNOTATIONS, ARGUMENTS AND REASONING*  
RT-MAC-9601, janeiro de 1996, 22 pp.

Kunio Okuda  
*REDUÇÃO DE DEPENDÊNCIA PARCIAL E REDUÇÃO DE DEPENDÊNCIA GENERALIZADA*  
RT-MAC-9602, fevereiro de 1996, 20 pp.

Junior Barrera, Edward R. Dougherty and Nina Sumiko Tomita  
*AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY DESIGN OF STATISTICALLY  
OPTIMAL OPERATORS IN THE CONTEXT OF COMPUTATIONAL LEARNING THEORY.*  
RT-MAC-9603, abril de 1996, 48 pp.

Junior Barrera e Guillermo Pablo Salas  
*SET OPERATIONS ON CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC  
PROGRAMMING OF MMACH'S*  
RT-MAC-9604, abril de 1995, 66 pp.

Kunio Okuda  
*CYCLE SHRINKING BY DEPENDENCE REDUCTION*  
RT-MAC-9605, maio de 1996, 25 pp.

Julio Stern, Fabio Nakano and Marcelo Lauretto  
*REAL- REAL ATTRIBUTE LEARNING FOR STRATEGIC MARKET OPERATION*  
RT-MAC-9606, agosto de 1996, 16 pp.

Markus Endler

*SISTEMAS OPERACIONAIS DISTRIBUÍDOS: CONCEITOS, EXEMPLOS E TENDÊNCIAS*

RT-MAC-9607, agosto de 1996, 120 pp.

Hae Yong Kim

*CONSTRUÇÃO RÁPIDA E AUTOMÁTICA DE OPERADORES MORFOLÓGICOS E EFICIENTES PELA*

*APRENDIZAGEM COMPUTACIONAL*

RT-MAC-9608, outubro de 1996, 19 pp.

Marcelo Finger

*NOTES ON COMPLEX COMBINATORS AND STRUCTURALLY-FREE THEOREM PROVING*

RT-MAC-9609, dezembro 1996, 28 pp.

Carlos Eduardo Ferreira, Flávio Keidi Miyazawa e Yoshiko Wakabayashi (eds)

*ANAIIS DA I OFICINA NACIONAL EM PROBLEMAS DE CORTE E EMPACOTAMENTO*

RT-MAC-9610, dezembro de 1996, 65 pp.

Carlos Eduardo Ferreira, C. C. de Souza e Yoshiko Wakabayashi

*REARRANGEMENT OF DNA FRAGMENTS: A BRANCH-AND-CUT ALGORITHM*

RT-MAC-9701, janeiro de 1997, 24 pp.

Marcelo Finger

*NOTES ON THE LOGICAL RECONSTRUCTION OF TEMPORAL DATABASES*

RT-MAC-9702, março de 1997, 36 pp.

Flávio S. Corrêa da Silva, Wamberlo W. Vasconcelos e David Robertson

*COOPERATION BETWEEN KNOWLEDGE BASED SYSTEMS*

RT-MAC-9703, abril de 1997, 18 pp.

Junior Barrera, Gerald Jean Francis Benon, Roberto de Alencar Lotufo, Roberto Hirata Junior

*MMACH: A MATHEMATICAL MORPHOLOGY TOOLBOX FOR THE KHOROS SYSTEM*

RT-MAC-9704, maio de 1997, 67 pp.

Julio Michael Stern e Cibele Dunder

*PORTFÓLIOS EFICIENTES INCLUINDO OPÇÕES*

RT-MAC-9705, maio de 1997, 29 pp.

Junior Barrera e Ronaldo Fumio Hashimoto

*COMPACT REPRESENTATION OF W-OPERATORS*

RT-MAC-9706, julho de 1997, 13 pp.

Dilma M. Silva e Markus Endler

*CONFIGURAÇÃO DINÂMICA DE SISTEMAS*

RT-MAC-9707, agosto de 1997, 35 pp

Kenji Koyama e Routh Terada

*AN AUGMENTED FAMILY OF CRYPTOGRAPHIC PARITY CIRCUITS*

RT-MAC-9708, setembro de 1997, 15 pp

Routh Terada e Jorge Nakahara Jr.

*LINEAR AND DIFFERENTIAL CRYPTANALYSIS OF FEAL-N WITH SWAPPING*

RT-MAC-9709, setembro de 1997, 16 pp

Flávio S. Corrêa da Silva e Yara M. Michelacci

*MAKING OF AN INTELLIGENT TUTORING SYSTEM (OR METHODOLOGICAL ISSUES OF ARTIFICIAL*

*INTELLIGENCE RESEARCH BY EXAMPLE)*

RT-MAC-9710, outubro de 1997, 16 pp.

Marcelo Finger

*COMPUTING LIST COMBINATOR SOLUTIONS FOR STRUCTURAL EQUATIONS*

RT-MAC-9711, outubro de 1997, 22 pp.

Maria Angela Gurgel and E.M.Rodrigues

*THE F-FACTOR PROBLEM*

RT-MAC-9712, dezembro de 1997, 22 pp.

Perry R. James, Markus Endler, Marie-Claude Gaudel

*DEVELOPMENT OF AN ATOMIC-BROADCAST PROTOCOL USING LOTOS*

RT-MAC-9713, dezembro de 1997, 27 pp.

Carlos Eduardo Ferreira and Marko Loperic

*A BRANCH-AND-CUT ALGORITHM FOR A VEHICLE ROUTING PROBLEM WITH CAPACITY AND TIME CONSTRAINTS*

RT-MAC-9714, dezembro de 1997, 20 pp.

Nami Kobayashi

*A HIERARCHY FOR THE RECOGNIZABLE M-SUBSETS*

RT-MAC-9715, dezembro de 1997, 47 pp.

Flávio Soares Corrêa da Silva e Daniela Vasconcelos Carbogim

*A TWO-SORTED INTERPRETATION FOR ANNOTATED LOGIC*

RT-MAC-9801, fevereiro de 1998, 17 pp.

Flávio Soares Corrêa da Silva, Wamberto Weber Vasconcelos, Jaume Agustí, David Robertson e Ana Cristina V. de Melo.

*WHY ONTOLOGIES ARE NOT ENOUGH FOR KNOWLEDGE SHARING*

RT-MAC-9802, outubro de 1998, 15 pp.

J. C. de Pina e J. Soares

*ON THE INTEGER CONE OF THE BASES OF A MATROID*

RT-MAC-9803, novembro de 1998, 16 pp.

K. Okuda and S.W.Song

*REVISITING HAMILTONIAN DECOMPOSITION OF THE HYPERCUBE*

RT-MAC-9804, dezembro 1998, 17pp.