

DMSr with Non-Precedence Constraints and Communication Tasks and Research of its Optimal Communication

Jun Li ^{1,+}, Yongfeng Diao ¹, Yajun Chen ¹ and Xing Yin ²

¹ School of Computer Science, China West Normal University, Nanchong 637002, China

² School of Computer Science, Pan Zhi Hua University, Panzhihua 637000, China

(Received June 26, 2008, accepted October 12, 2008)

Abstract. Firstly, the distributed communication task model with non-precedence constraints is described. Based on the DMSr, distributed multimedia Sr(DMSr) algorithm with non-precedence constraints and communication tasks(DMSrNPCC) is preliminarily studied. Secondly, the optimum communication problem of DMSrNPCC is described. This problem is changed into the map division, and based on heuristic algorithm(MMP – Slover) a solution is suggested, which gradually optimizes the division from communication and judge whether sacrificing the division for meeting the scheduling condition, beginning with any division.

Keywords: non-independent tasks; DMSr algorithm; heuristic algorithm, graphical partition

1. Introduction

Pinwheel scheduling model is a kind of Distance Constraints Task System(DCTS), which is the temporal distance between any two adjacent executions of the same task should always be less than a certain amount of time. The advantage of DCTS is the schedule with jitterless by curtailing appropriately the scheduling period on condition that the delay of task period is guaranteed. Ching-Chih Han introduced Pinwheel scheduling algorithm for a single node, Sr[1]. Chih-Wen Hsueh extended Sr to more than one nodes and introduced the algorithm of distributed Sr(DSr)[2]. DSr had two tolerance requests that the running orders of all tasks had same node order in distributed system and every task had the same distance constraints in node. So, DSr was expended into distributed multimedia tasks, and Zhang introduced a jitterless integrated task scheduling algorithm DMSr[3]. But all tasks are independent in Sr, DSr and DMSr. In really, there are communication costs among tasks. We research the schedule of non-precedence constraints and communication tasks(NPCCT), and give the running time of task an increment, which is communication cost. Then communication task is considered to independent task. Based on DMSr with independent tasks, we preliminarily study DMSr with non-precedence constraints and communication tasks(DMSrNPCC). In distributed system tasks have out-node communication cost(OCC) and in-node communication cost(ICC), which is neglected because it is a little, Zheng introduced the heuristic algorithm MMP-Slover that reduce the whole communication cost[4]. In MMP-Slover the number of every node is uncertainty, but the number of every node is certainty and task must be schedulable in DMSr. In order to optimize the communication cost of DMSr, then a constraint MMP-Slover algorithm is proposed. We use the constraint MMP-Slover to optimize the communication cost among tasks, which have more OCC, and reduce the whole communication cost.

This paper has the following organization. In this section (Section 1), we provide the introduction. In Section 2, we describe the distributed communication tasks model of non-precedence constraints and study the NPCCDMSr. In Section 3, we research the optimum communication problem of NPCCDMSr and suggest a solution. In Section 4, we give an example. The last section offers the conclusion of this paper.

2. DMSr with non-precedence constraints and communication tasks

Firstly, we describe the non-precedence constraints and communication tasks.

⁺ Corresponding author. Tel.: +86-0817-2184805; Fax: +86-0817-2568032.
E-mail address: lj_1202@126.com.

In distributed system $N = \{N_j | 1 \leq j \leq m\}$ is node set. We suppose that all nodes have the same ability. $X = \{X_i | 1 \leq i \leq n\}$ is the distance constraints task set of NPCCT, and $\forall X_i \in X, X_i = \{T_{ij}\}$, where T_{ij} is a task that task X_i is running in node N_j . T_{gh} is a task that task X_g is running in node $N_h, 1 \leq g \leq n, 1 \leq h \leq m$. When $j \neq h$, T_{ij} and T_{gh} are not in a node and OCC is $U_{T_{ij}, T_{gh}}$. When $j = h$, T_{ij} and T_{gh} are in a node and ICC is $U'_{T_{ij}, T_{gh}}$. Because in-node communication cost is a little ($U'_{T_{ij}, T_{gh}} \ll U_{T_{ij}, T_{gh}}$), $U'_{T_{ij}, T_{gh}}$ is neglected and communication cost only is OCC[4]. Communication cost matrix between T_{ij} and other tasks is denoted by $\Psi_{ij} = \{U_{T_{ij}, T_{gh}} | 1 \leq g \leq n, 1 \leq h \leq m\}$.

Communication cost from T_{ij} to T_{gh} is same to from T_{gh} to T_{ij} . In order to analyze, $U_{T_{ij}, T_{gh}}$ is OCC in this paper, when $j = h$. For all nodes have the same ability, allocating task dose not concern the node. Task T_{ij} has the running time d_{ij} and distance constraint c_{ij} . The running really time e_{ij} of task T_{ij} , which is the sum of the running time and the communication cost, is denoted by $e_{ij} = d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}}$, where $U_{T_{ij}, T_{gh}} \in \Psi_{ij}$. The expression indicates that the running really time is minimal when $\sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}}$ is the minimum value.

The running really time e_{ij} of this paper is different from e_{ij} of reference 3, so running really time matrix E is different from E of reference 3. The running really time matrix E , the distance constraint matrix C , the scheduling period matrix B and the routing matrix R are defined and are denoted by

$$E = \begin{bmatrix} e_{11}, e_{12}, \dots, e_{1m} \\ e_{21}, e_{22}, \dots, e_{2m} \\ \dots \\ e_{n1}, e_{n2}, \dots, e_{nm} \end{bmatrix} \quad C = \begin{bmatrix} c_{11}, c_{12}, \dots, c_{1m} \\ c_{21}, c_{22}, \dots, c_{2m} \\ \dots \\ c_{n1}, c_{n2}, \dots, c_{nm} \end{bmatrix} \quad B = \begin{bmatrix} B_{11}, B_{12}, \dots, B_{1m} \\ B_{21}, B_{22}, \dots, B_{2m} \\ \dots \\ B_{n1}, B_{n2}, \dots, B_{nm} \end{bmatrix} \quad R = \begin{bmatrix} r_{11}, r_{12}, \dots, r_{1m} \\ r_{21}, r_{22}, \dots, r_{2m} \\ \dots \\ r_{n1}, r_{n2}, \dots, r_{nm} \end{bmatrix}$$

Rows of matrixes E , C and B are running really time, distance constraint and schedule period of every task in every node respectively. Columns of matrixes E , C and B are running really time, distance constraint and scheduling period of all tasks in every node respectively. To describe that different tasks are in different nodes, Zhang defined the routing matrix R . If task X_i is in j^{th} node, $r_{ij} = 1$, else $r_{ij} = 0$. If $r_{ij} = 0$, all parameters of task T_{ij} are 0.

Definition 1[3]: $\forall k, l_k = c_k / 2^{\lceil \log(c_k/c_{\min}) \rceil}$ is the partition basis of c_k , where c_{\min} is the minimum value of c .

Definition 2[3]: Let a DCTS be a distributed system that have m nodes and n tasks. The density of node j is $\rho_j(l) = \sum_{i=1}^n \frac{e_{ij} \cdot r_{ij}}{B_{ij}}$, where l is the partition basis of schedule period; The density of system is

$$\Phi = \sum_{j=1}^m \rho_j(l).$$

We propose DMSr with non-precedence constraints and communication tasks that we apply DMSr algorithm based on running really time matrix E , as follow:

First step: calculating the real running time matrix E , where $e_{ij} = d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}}$;

Second step: DMSr algorithm[3].

1. We compute the following steps in parallel and select candidates of partition basis in every node.

1.1 To efficient task of distance constraint $C_j = (c_{1j}, c_{1j}, \dots, c_{nj})^T$ of the element node, we calculate:

$$l_{ij} = c_{ij} / 2^{\lfloor \log(c_{ij}/c_{\min}) \rfloor}, \text{ where } c_{\min} \text{ is the minimum value of } C_j, \text{ we obtain } L_j = (l_{1j}, l_{2j}, \dots, l_{nj})^T;$$

1.2 $\forall i$, with l_{ij} , C_j is divided. We calculate the density: $B_{ik} = l_{ij} \cdot 2^{\lfloor \log(c_{ik}/l_{ij}) \rfloor}, 1 \leq i \leq n, 1 \leq k \leq n$, and

$$\text{have: } B = \begin{bmatrix} B_{11}, B_{12}, \dots, B_{1n} \\ B_{21}, B_{22}, \dots, B_{2n} \\ \dots\dots\dots \\ B_{n1}, B_{n2}, \dots, B_{nm} \end{bmatrix}$$

1.3 With the running really time $E_j = (e_{1j}, e_{2j}, \dots, e_{nj})^T$ and the routing matrix $R_j = (r_{1j}, r_{2j}, \dots, r_{nj})^T$ of the element node, we calculate the density of the element node

$$\rho_{ij} = \sum_{k=1}^n (e_{kj} \cdot r_{kj} / B_{ik}), 1 \leq i \leq n, \text{ then have: } \rho_j = (\rho_{1j}, \rho_{2j}, \dots, \rho_{nj})^T.$$

1.4 $S_j = (s_{1j}, s_{2j}, \dots, s_{nj})^T$ is defined. If and only if l_{ij} is valid, $s_{ij} = 1$, else $s_{ij} = 0$. $\forall i$, if $s_{ij} = 1$ and $\rho_{ij} \leq 1$, $s_{ij} = 1$, else $s_{ij} = 0$. Then select candidates of partition basis.

1.5 Candidate is tested in $N_1, \dots, N_{j-1}, N_{j+1}, \dots, N_m$.

1.5.1 The distance constraint of node $B'_{ij} = l_{ij} \cdot 2^{\lfloor \log(c'_{ij}/l_{ij}) \rfloor}, 1 \leq i' \leq n, 1 \leq j' \leq n \wedge j' \neq j$ is calculated,

$$\text{and we obtain: } B = \begin{bmatrix} B'_{11}, \dots, B'_{1(j-1)}, B'_{1(j+1)}, \dots, B'_{1m} \\ B'_{21}, \dots, B'_{2(j-1)}, B'_{2(j+1)}, \dots, B'_{2m} \\ \dots\dots\dots \\ B'_{n1}, \dots, B'_{n(j-1)}, B'_{n(j+1)}, \dots, B'_{nm} \end{bmatrix}, \text{ where the element that value is 1 in } S_j \text{ is the}$$

corresponding element that is candidate of partition basis in L_j .

1.5.2 We calculate the density of node $\rho'_{ij} = \sum_{k=1}^n (e_{kj} \cdot r_{kj} / B'_{ik}), 1 \leq j' \leq m \wedge j' \neq j$, and have:

$$\rho_i = (\rho_{i1}, \dots, \rho_{i(j-1)}, \rho_{i(j+1)}, \dots, \rho_{im}).$$

1.5.3 If $\rho_{i1} \leq 1 \wedge \rho_{i2} \leq 1 \wedge \dots \wedge \rho_{im} \leq 1 \wedge j' \neq j$ in ρ_j , s_{ij} is unchanging, else $s_{ij} = 0$.

1.6 Via 1.5, we obtain the partition basis L_j , S_j and the density of system $\rho_j = \begin{bmatrix} \rho_{11}, \rho_{12}, \dots, \rho_{1m} \\ \rho_{21}, \rho_{22}, \dots, \rho_{2m} \\ \dots\dots\dots \\ \rho_{n1}, \rho_{n2}, \dots, \rho_{nm} \end{bmatrix}$,

calculate $\Phi_{ij} = \sum_{k=1}^m \rho_{ik} \cdot s_i$, and have $\Phi_j = (\Phi_{1j}, \Phi_{2j}, \dots, \Phi_{nj})$.

2. We select the partition basis of system in key node of distributed system.

2.1 We find the minimum value in Φ that the corresponding element of S is 1, then the corresponding element of L serves as the partition basis l^* .

2.2 All tasks are parted, that $B_{ij} = l^* \cdot 2^{\left\lfloor \log \left(\frac{e_{ij} r_{ij}}{l^*} \right) \right\rfloor}$, $1 \leq i \leq n, 1 \leq j \leq m$. Then we obtain the scheduling period matrix B of every node.

Apparently, time complexity of NPCCDMSr is $O(m^2 n^2)$, because time complexity of first step is $O(m^2 n^2)$, and time complexity of DMSr is $O(mn^2)$ [3].

3. Scheduling Condition of DMSrNPCC and its optimal communication

Zheng considered the whole parallel running really time $e = \text{Max} \{e_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$. Optimal aim reduces out-node communication cost, that is the minimum of e . A MMP-Slover algorithm is proposed in reference 4. But optimal communication tasks are not real-time scheduling tasks. This paper analyzes the optimal communication of DMSrNPCC based on MMP-Slover algorithm. Firstly, we study scheduling condition of tasks in DMSrNPCC algorithm.

3.1. Scheduling Condition of DMSrNPCC

Definition 3[3]: Let X be a distributed and scheduling multimedia task set in DCTS. $\forall j (1 \leq j \leq m)$,

$\tau_j(X) = \sum_{i=1}^n \frac{e_{ij} \cdot r_{ij}}{c_{ij}}$ is the density of distance constraint that X is in node N_j .

Lemma 1[3]: Let X be a distributed and scheduling multimedia task set in DCTS. In arbitrary node N_j , if task set X is satisfied with $\tau_j(X) \leq \beta = 2^{1/n_j - 1}$, where n_j is the number of task in node N_j , then X is schedulable in DCTS.

When task T_{ij} is given, the running time d_{ij} and distance constraint c_{ij} are certainty. When all tasks are given, $U_{T_{ij}, T_{gh}}, r_{ij}$ and n_j are certainty too.

Definition 4: Let $X = \{X_i\}$ be a distributed and scheduling task set in NPCCDCTS, where $X_i = \{T_{ij}\}$,

$1 \leq i \leq n, 1 \leq j \leq m. \forall j (1 \leq j \leq m), \delta_j(X) = \sum_{i=1}^n \left(d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}} \right) \cdot r_{ij} / c_{ij}$ is the density of distance constraint that X is in node N_j .

Thermo 1: Let $X = \{X_i\}$ be a distributed and scheduling task set in NPCCDCTS, where $X_i = \{T_{ij}\}$, $1 \leq i \leq n, 1 \leq j \leq m$. In arbitrary node N_j , if task set X is satisfied with $\delta_j(X) \leq 2^{1/n_j - 1}$, where n_j is the number of task in node N_j , then X is schedulable in NPCCDCTS.

Proof : Let $X = \{X_i\}$ be a distributed and scheduling task set in NPCCDCTS. $d_{ij}, c_{ij}, \Psi_{ij}, r_{ij}$ and n_j are certainty. We calculate the running really time of T_{ij} by $e_{ij} = d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}}$. Based on definition 2,

$\forall j, \delta_j(X) = \sum_{i=1}^n \left(d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}} \right) \cdot r_{ij} / c_{ij}$ is the density of distance constraint that X is in node N_j .

Based on lemma 1, In arbitrary node N_j , If task X is satisfied with $\delta_j(X) \leq 2^{1/n_j - 1}$, where n_j is the number of task in node N_j , then X is schedulable in NPCCDCTS.

3.2. Describing Optimal Communication Problem

Because communication cost among tasks exists, we do our best to reduce communication cost and increase efficiency of nodes. For the sake of balanced load of nodes, we consider $e = \text{Max}\{e_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$. Suppose that the running time of all tasks exists, optimal aim

translates into optimal communication, namely: minimum of $\text{Max}\left(\sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}} \mid 1 \leq i \leq n, 1 \leq j \leq m\right)$.

But optimal prerequisites are that tasks are schedulable. Optimal communication is describing as:

Objective function:

$$Z_{Min} = \text{Max}_{1 \leq i \leq n, 1 \leq j \leq m} \left(\sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}} \right) \dots\dots\dots (*)$$

Optimal conditions:

$$\delta_j(X) = \sum_{i=1}^n \left(d_{ij} + \sum_{g=1}^n \sum_{h=1, \neq j}^m U_{T_{ij}, T_{gh}} \right) \cdot r_{ij} / c_{ij} \leq 2^{1/n_j - 1}, 1 \leq j \leq m \dots\dots\dots (\star)$$

All tasks represent nodes of graph. And out-node communication costs of tasks represent weight of connected edges in the graph. We convert optimal communication into partition problem of graph.

Definition 5[4]: Let $G = (V, F)$ be a graph, where $V = \{v_1, v_2, \dots, v_{nm}\}$ and $F = \{f_1, f_2, \dots, f_b\}$. $S(v_i)$ is capacity of node v_i , which is the running time of task. $W(f_j)$ is weight of f_j . We suppose that all capacities and running times are nonnegative numbers. $S(V)$ is capacity of node set V , where $S(V) = \sum_{v \in V} S(v)$.

1) In schedulable conditions of formula (\star), G is divided into P_1, P_2, \dots, P_m by partition Y . When $\sum_{1 \leq j < h \leq m} |S(P_j) - S(P_h)|$ is minimum value, Y is called balanceable partition.

2) $W_s(P_j, P_h)$ is the sum of weights between P_j and P_h , where $W_s(P_j, P_h) = \sum_{g \in P_j, h \in P_h} W((g, h))$. $S_{ws}(P_j)$ is the sum of weights between P_j and all other subsets, where $S_{ws}(P_j) = \sum_{1 \leq h \leq m, j \neq h} W_s(P_j, P_h)$.

3) Objective function: $Y_s(Y) = \sum_{1 \leq j < k} S_{ws}^3(P_j)$ and $T_{ys}(Y) = \sum_{1 \leq j < h \leq k} S(P_j) \cdot S(P_h) \cdot \xi - Y_s(Y)$, where $\xi (\xi \in \mathbb{Z}^+)$ is a constant, where $\xi > \left(\sum_{j=1}^{|F|} W(f_j) \right)^3$.

4) When $v_r (v_r \in P_{from})$ swaps with $v_d (v_d \in P_{to})$, $g(v_r, v_d)$ is the variation of $T_{ys}(Y)$, where $g(v_r, v_d) = T_{ys}(Y^+) - T_{ys}(Y)$, in which Y is the preceding partition and Y^+ is a new partition after v_r swaps with v_d .

5) $W_{n2p}(v_r, P_j)$ is the sum of weights between node v_r and subset P_j , where $W_{n2p}(v_r, P_j) = \sum_{u \in P_j} W((v_r, u))$. $W'_{n2p}(v_r, P_j)$ is the sum of weights between node v_r and all subsets except P_j , where $W'_{n2p}(v_r, P_j) = \sum_{u \notin P_j} W((v_r, u))$.

So, optimal communication is converted into TMCSPP[4] with conditions of formula (\star), which is

called NPCCTMCSP.

3.3. Optimal Communication of DMSrNPCC

We propose a solution that starting from partition Y , Y is optimized step by step and judged the scheduling condition. The algorithm thought is as follow:

1. We suppose that starting from Y , current Y and interrelated information, which consists of the running time d_{ij} , distance constraint c_{ij} , communication cost Ψ_{ij} , the routing r_{ij} and the scheduling condition, save as Y° .

2. Starting from Y° , firstly, we choose two tasks T_{ij} ($T_{ij} \in P_j$) and T_{gh} ($T_{gh} \in P_h$) that T_{ij} swaps with T_{gh} , $1 \leq j, h \leq m, j \neq h$. If the scheduling condition of formula (★) is not satisfied (it only judges the scheduling condition of N_j and N_h , others may be judged by the save value.), turn to 2, until all tasks swap once.

3. If the variation of $T_{ys}(Y)$ does not increase or the increment of $T_{ys}(Y)$ is small, we go on, else we choose the swapped task that the increment of $T_{ys}(Y)$ is the maximum, and do that T_{ij} swaps with T_{gh} (d_{ij} swaps with d_{gh} , c_{ij} swaps with c_{gh} , Ψ_{ij} swaps with Ψ_{gh} , r_{ij} swaps with r_{gh} and we replace Y° with current Y), simultaneously turn to 2.

4. If Y° is schedulable, Y can be optimal schedule, else Y can be not optimal schedule.

5. We start from other partition Y' , then turn to 2.

6. Over.

Apparently, time complexity of the solution is $O(n^2 m^3)$.

A key of the solution is the node and task subset that we choose. So, the variation of $T_{ys}(Y)$ must be calculated. Next, we prove two Thermos.

Lemma 2[4]: When node v_r is moved from P_{from} to P_{to} , we have:

$$\begin{aligned} S_{ws}(P_{from}^*) &= S_{ws}(P_{from}) + W_{n2p}(v_r, P_{from}) - W'_{n2p}(v_r, P_{from}) \\ S_{ws}(P_{to}^*) &= S_{ws}(P_{to}) - W_{n2p}(v_r, P_{to}) + W'_{n2p}(v_r, P_{to}) \end{aligned}$$

Where P_i^* ($1 \leq i \leq m$) is new task subset after movement.

Lemma 3[4]: When node v_r is moved from P_{from} to P_{to} , we have:

$$g(v_r, P_{to}) = (S(P_{from}) - S(P_{to}) - 1)\xi + (S_{ws}^3(P_{from}) + S_{ws}^3(P_{to}) - S_{ws}^3(P_{from}^*) - S_{ws}^3(P_{to}^*)).$$

Thermo 2: When v_r swaps with v_d , $v_r \in P_{from}$, $v_d \in P_{to}$, we have:

$$\begin{aligned} S_{ws}(P_{from}^+) &= S_{ws}(P_{from}) + W_{n2p}(v_r, P_{from}) - W'_{n2p}(v_r, P_{from}) - W_{n2p}(v_d, P_{from}) + W'_{n2p}(v_d, P_{from}) + 2U_{v_r, v_d} \\ S_{ws}(P_{to}^+) &= S_{ws}(P_{to}) - W_{n2p}(v_r, P_{to}) + W'_{n2p}(v_r, P_{to}) + W_{n2p}(v_d, P_{to}) - W'_{n2p}(v_d, P_{to}) + 2U_{v_r, v_d} \end{aligned}$$

Where P_i^+ ($1 \leq i \leq m$) is new task subset after swap.

Proof: firstly, node v_r is moved from P_{from} to P_{to} , P_{from} turns into P_{from}^* , P_{to} turns into P_{to}^* , by Lemma 2 we have:

$$\begin{aligned} S_{ws}(P_{from}^*) &= S_{ws}(P_{from}) + W_{n2p}(v_r, P_{from}) - W'_{n2p}(v_r, P_{from}) \\ S_{ws}(P_{to}^*) &= S_{ws}(P_{to}) - W_{n2p}(v_r, P_{to}) + W'_{n2p}(v_r, P_{to}) \end{aligned}$$

Secondly, node v_d is moved from P_{to}^* to P_{from}^* , P_{from}^* turns into P_{from}^+ , P_{to}^* turns into P_{to}^+ , by Lemma 2 we have:

$$S_{ws}(P_{from}^+) = S_{ws}(P_{from}^*) - W_{n2p}(v_d, P_{from}^*) + W'_{n2p}(v_d, P_{from}^*)$$

$$S_{ws}(P_{to}^+) = S_{ws}(P_{to}^*) + W_{n2p}(v_d, P_{to}^*) - W'_{n2p}(v_d, P_{to}^*)$$

Because of $W_{n2p}(v_d, P_{to}^*) = W_{n2p}(v_d, P_{to}) + U_{v_r, v_d}$, $W'_{n2p}(v_d, P_{to}^*) = W'_{n2p}(v_d, P_{to}) - U_{v_r, v_d}$,

$$W_{n2p}(v_d, P_{from}^*) = W_{n2p}(v_d, P_{from}) - U_{v_r, v_d}, W'_{n2p}(v_d, P_{from}^*) = W'_{n2p}(v_d, P_{from}) + U_{v_r, v_d}$$

We have the result by substitution method.

Thermo 3: When v_r swaps with v_d , $v_r \in P_{from}$, $v_d \in P_{to}$, we have:

$$g(v_r, v_d) = 2(d_{v_r} - d_{v_d} - 1)\xi + S_{ws}^3(P_{from}) + S_{ws}^3(P_{to}) - S_{ws}^3(P_{from}^+) - S_{ws}^3(P_{to}^+)$$

Proof : firstly, node v_r is moved from P_{from} to P_{to} , P_{from} turns into P_{from}^* , P_{to} turns into P_{to}^* , by Lemma 3

we have: $g(v_r, P_{to}) = (S(P_{from}) - S(P_{to}) - 1)\xi + (S_{ws}^3(P_{from}) + S_{ws}^3(P_{to}) - S_{ws}^3(P_{from}^*) - S_{ws}^3(P_{to}^*))$

Secondly, node v_d is moved from P_{to}^* to P_{from}^* , P_{from}^* turns into P_{from}^+ , P_{to}^* turns into P_{to}^+ , by Lemma 3 we

have: $g(v_d, P_{from}^*) = (S(P_{to}^*) - S(P_{from}^*) - 1)\xi + (S_{ws}^3(P_{from}^*) + S_{ws}^3(P_{to}^*) - S_{ws}^3(P_{from}^+) - S_{ws}^3(P_{to}^+))$

So, $g(v_r, v_d) = g(v_r, P_{to}) + g(v_d, P_{from}^*)$. The running time of nodes v_r and v_d are respectively d_{v_r} and d_{v_d} . Then $S(P_{to}^*) = S(P_{to}) + d_{v_r} - d_{v_d}$, $S(P_{from}^*) = S(P_{from}) - d_{v_r} + d_{v_d}$. We have the result by substitution method.

4. Example

We introduce the optimal communication by an example. T_{11} and T_{12} are in node N_1 ; T_{21} , T_{22} and T_{23} are in node N_2 ; T_{31} and T_{33} are in node N_3 ; T_{42} and T_{43} are in node N_4 from table 1. There are the running time, the running really time and constraint time in table 1.

Table 1 task set

node	task	Running time (ms)	Running really time (ms)	constraint time(ms)
N_1	T_{11}	1	1.5	9.2
	T_{12}	1.4	1.4	8.9
N_2	T_{21}	5.1	5.1	25
	T_{22}	0.9	1.4	9.1
	T_{23}	3.8	7.2	30.7
N_3	T_{31}	7.7	7.7	30.8
	T_{33}	6.4	9.8	30.6
N_4	T_{42}	2.3	2.3	20.1
	T_{43}	1.7	1.7	22.5

We know that all tasks are schedulable by definition 4 and there are 7.8(ms) of communication cost from table 2. So, we are concerned with reducing the communication cost as much as possible. Via optimal communication algorithm, we obtain that firstly T_{11} swaps with T_{21} and secondly T_{23} swaps with T_{31} . Then the communication cost time is zero and all tasks are schedulable in nodes. T_{21} and T_{12} are in node N_1 ; T_{11} , T_{22} and T_{31} are in node N_2 ; T_{23} and T_{33} are in node N_3 ; T_{42} and T_{43} are in node N_4 .

Table 2 communication cost set

		Ψ_{11}	Ψ_{12}	Ψ_{13}	Ψ_{21}	Ψ_{22}	Ψ_{23}	Ψ_{31}	Ψ_{32}	Ψ_{33}	Ψ_{41}	Ψ_{42}	Ψ_{43}
		T_{11}	T_{12}		T_{21}	T_{22}	T_{23}	T_{31}		T_{33}		T_{42}	T_{43}
Ψ_{11}	T_{11}	0	0	0	0	0.5	0	0	0	0	0	0	0
Ψ_{12}	T_{12}	0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{13}		0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{21}	T_{21}	0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{22}	T_{22}	0.5	0	0	0	0	0	0	0	0	0	0	0
Ψ_{23}	T_{23}	0	0	0	0	0	0	0	0	3.4	0	0	0
Ψ_{31}	T_{31}	0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{32}		0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{33}	T_{33}	0	0	0	0	0	3.4	0	0	0	0	0	0
Ψ_{41}		0	0	0	0	0	0	0	0	0	0	0	0
Ψ_{42}	T_{42}	0	0	0	0	0	0	0	0	0.9	0	0	0
Ψ_{43}	T_{43}	0	0	0	0	0	0	0	0	0	0	0	0
	Sum	0.5	0	0	0	0.5	3.4	0	0	3.4	0	0	0

5. Conclusion

This paper gives communication cost matrix Ψ_{ij} between T_{ij} and other tasks. We add communication cost to running time d_{ij} , and obtain the real running time e_{ij} . Based on the DMSr algorithm, we study DMSr with communication and non-precedence constraints tasks, analyze scheduling condition of DMSrNPCC, and describe the problem of optimal communication of DMSrNPCC. Finally, we convert optimal communication into partition problem of graph, and propose a solution.

6. Acknowledgments

This work was supported by the Important Project Function of Science and Technology Bureau of Sichuan Province of China (Grant NO. 05JY029-093).

7. References

- [1] Ching-Chih Han, Kwei-Jay Lin. Scheduling Distance-Constrained Real-Time Tasks. *Proceeding of IEEE Real-Time System Symposium*. 1992, pp. 300-308. New York.
- [2] Ching-Chih Han, Kwei-Jay Lin, Fan Nong. Distributed Pinwheel Scheduling with end-to-end Timing Constraints. *Proceeding of IEEE Real-Time System Symposium*. 1995, pp. 172-181. Pisa.
- [3] Zhang Zhanjun, Yang Xueliang. A Jitterless tasks scheduling algorithm for distributed multimedia systems. *Chinese Journal of Computers*. 1999, **21**(1): 24-30. (in Chinese)
- [4] Zheng weiming, Yang Bo, Lin Weijian, etal. Scheduling parallel tasks of optimal communication in SMP cluster. *Science in China Series E: Technological Sciences*. 2001, **31**(5): 442-454. (in Chinese)