Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 723497, 7 pages
http://dx.doi.org/10.1155/2014/723497

Research Article

Complexity Analysis of New Task Allocation Problem Using Network Flow Method on Multicore Clusters

Jixiang Yang

School of Science, Chongqing Jiaotong University, Chongqing 400074, China

Correspondence should be addressed to Jixiang Yang; jixiang_yang@126.com

Received 9 October 2013; Accepted 18 December 2013; Published 2 January 2014

Academic Editor: Balaji Raghavan

The task allocation problem (TAP) generally aims to minimize total execution cost and internode communication cost in traditional parallel computing systems. New TAP (NTAP) considering additive intranode communication cost in emerging multicore cluster systems is investigated in this paper. We analyze the complexity of NTAP with network flow method and conclude that the intranode communication cost is a key to the complexity of NTAP, and prove that (1) the NTAP can be cast as a generalized linear network minimum cost flow problem and can be solved in $O(m^2n^4)$ time if the intranode communication cost equals the internode communication cost, and (2) the NTAP can be cast as a generalized convex cost network minimum cost flow problem and can be solved in polynomial time if the intranode communication cost is more than the internode communication cost. More in particular, the uniform cost NTAP can be cast as a convex cost flow problem and can be solved in $O(m^2n^2\log(m+n))$ time. Furthermore, solutions to the NTAP are also discussed. Our work extends currently known theoretical results and the theorems and conclusions presented in this paper can provide theoretical basis for task allocating strategies on multicore clusters.

1. Introduction

Since the single core processors rapidly reach the physical limits of possible complexity and speed, computer architects have designed multicore processor, which means place two or more processing cores on the same chip. Multicore processors are now growing as a new industry trend and widely used for high performance computing. Further, multicore processors are being configured in a hierarchical manner to compose computing nodes or multicore nodes in cluster systems. Multicore clusters based on these computing nodes or multicore nodes have already been one of the most popular models in parallel computing [1, 2].

However, for a multicore node, on one side, the future performance growth in multicore processors will almost certainly come from the exploitation of thread-level parallelism through multicore processors, which consequently can lead to memory access contention when multiple cores concurrently access the shared resources such as memory, cache, and disk I/O. The synchronization operation introduced to avoid the access contention can require a lot of overhead. In a larger-scale multicore node or high-contention situations, synchronization can become a performance bottleneck because contention introduces additional delays and because latency is potentially greater in such a multicore computing node. On the other side, from the message distribution experiments, it is found that on an average, about 50% messages are transferred through intranode communication, which is much higher than intuition. This trend indicates that considering the intranode communication is as important as considering the internode communication on a multicore cluster [1]. As a matter of fact, synchronization can be considered as a special form of communication [3]. Therefore, in this paper, in order to facilitate description, the intranode communication overhead and synchronization overhead on a multicore node can be referred to as intranode communication cost. The intranode communication cost tends to increase dramatically when the numbers of multicore processors and tasks communicating on a multicore computing node increase. A report from Berkeley [4] predicts multicore processors with thousands of parallel execution units as the mainstream hardware of the future. Thus, the intranode communication cost has become a key factor to be considered in the TAP on multicore clusters.

In traditional parallel computing systems, the task allocation problem (TAP) is to assign a set of tasks or modules to a set of processors or computing nodes, so that the total execution cost and internode communication cost can be minimized [5–10]. To our best knowledge, new TAP considering overall execution cost, internode communication cost, and intranode communication cost in emerging multicore cluster systems has yet to be investigated. This paper proposes the new TAP (NTAP) aiming to minimize the total execution cost, internode communication cost, and intranode communication cost on multicore clusters. However, we are now encountering two important and challenging theoretical problems: (1) how can the complexity of the NTAP be efficiently analyzed and (2) what are the effects of intranode communication cost on the complexity of the NTAP. Aiming at the two important theoretical problems, we analyze and prove the effects of the intranode communication cost on the complexity of the NTAP via constructing equivalence relation between the NTAP and minimum cost flow problem. Moreover, solutions to the NTAP due to different complexity are also discussed.

The rest of this paper is organized as follows. After describing related work in Section 2, some basic definitions are provided in Section 3. Complexity analysis of the NTAP is performed in Section 4. Solutions to the NTAP are also discussed in Section 5. We conclude this paper in Section 6.

2. Related Work

TAP is a classical problem in the field of parallel computing research. Solution methods already suggested for this problem can be roughly classified into three categories [5], namely, graph theoretic approach, mathematical programming approach, and heuristic approach. The graph theoretic approach uses a graph to represent the interconnections between modules and represents the tasks to be allocated as a set of nodes or vertices of a graph. The intermodular communication cost between each pair of tasks is represented by the weight of a nondirected arc or a nondirected edge connecting two nodes or vertices. A communication cost of zero means that there is no communication between tasks or computing nodes, while a communication cost of infinity indicates that the communicating nodes or vertices must be assigned to the same processor or computing node. The mathematical programming approach formulates task assignment as an optimization problem and solves it with mathematical programming techniques. And the heuristic method provides fast but suboptimal algorithms for task assignment, which are useful for applications where an optimal solution cannot be obtained in real time.

In this paper, it is worth noting that our work is closely related to the graph theoretic approach, and our emphasis will be on the network flow method which is one of important graph theoretic approaches. For network flow method, each task and processor are represented by a node or a vertex. The network flow model can be built according to interconnections between modules, interprocessor communication, and task execution overhead on processor and can be solved with maximum flow and minimum cut algorithm. Research by Stone [6] and Bokhari [7] has shown how an optimal assignment may be found efficiently for the case of dual processor systems using a network flow algorithm. While an extension to three processors was developed by Stone [8], algorithms for four or more processors have not been found. Bokhari [9] has shown that the problem of finding an optimal assignment for four or more processors is a NP-complete problem and that the case where the graph of the communicating tasks, which we call communication graph, is a tree and can be solved exactly using dynamic programming. Towsley [10] generalized Bokhari's results to the case of series-parallel structures. From the theoretical point of view, by combining Bokhari's and Towsley's work, Fernandez-Baca [11] proposed polynomial time optimal algorithms in the case where the intertask communication graph is a k-tree. Lee et al. [12] and Cho and Park [13] have suggested optimal algorithms for the general structure problem in a linear array network with any number of processors. Fernandez de la Vega and Lamari [14] have investigated the case where all the tasks communicate with communication costs all equal to a constant c_0 and gave two exact polynomial time algorithms and a polynomial time approximation scheme using minimum cost flow theory. In addition, the problem of finding an optimal dynamic assignment of a modular program for a two-processor system is analyzed and Stone's formulation of the static assignment problem is extended to include the cost of dynamically reassigning a module from one processor to the other and the cost of module residence without execution by Bokhari [7]. Yadav et al. [15] have extended this model and considered the dynamic TAP for a general program structure and heterogeneous N processors in distributed computing systems.

Traditional TAP generally aims to minimize the total execution cost and internode communication cost without considering the intranode communication cost in multicore cluster computing, which frequently results in inefficient solutions since it cannot characterize and explore the hierarchical design features and potential of multicore clusters. Compared with above-mentioned traditional TAP, the NTAP considers the additive intranode communication cost and can fully characterize and exploit the hierarchical design features and potential of multicore clusters but still remains to be studied.

3. Preliminaries

Without loss of generality, let $T = \{t_1, t_2, \ldots, t_n\}$ be a set of n tasks and let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of m computing nodes. Let us denote a task assignment by a vector $Z = (z_1, z_2, \ldots, z_n)\{1, 2, \ldots, m\}^n$ and denote the total cost of an assignment by $C(Z)$, where $z_i = q$ means that t_i is allocated to p_q with $1 \le i \le n$ and $1 \le q \le m$. If a task assignment can minimize total execution cost, internode communication cost, and intranode communication cost, then we call it an optimal task assignment. Let x_q be the number of tasks assigned to p_q and let e_{iq} be the execution cost of t_i on p_q. Let the binary variable x_{iq} satisfy $x_{iq} \in \{0, 1\}$ and the x_{iq} is defined to be 1 if t_i is assigned to p_q and be 0 otherwise. Let the triple variable x_{ijq} satisfy $x_{ijq} \in \{0, 1, 2\}$ and the x_{ijq} is defined to be (1) 0 if t_i and t_j both are not allocated to p_q, (2) 1 if t_i or t_j is allocated to p_q, and (3) 2 if t_i and t_j are both allocated to p_q, where $1 \le j \le n$.

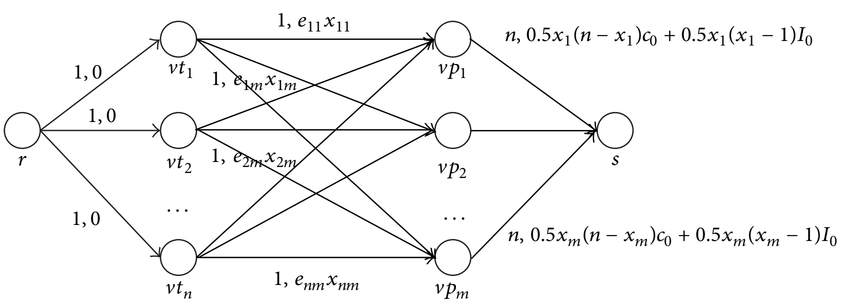


Figure 1: The MCF problem equivalent to UCNTAP.

Let c_{ij} denote the internode communication cost incurred between t_i and t_j assigned to distinct computing nodes and I_{ij} denote the intranode communication cost incurred between t_i and t_j allocated to the same computing node. We assume that $c_{ij} = 0$ if $z_i = z_j$, $I_{ij} = 0$ if $z_i \neq z_j$, and $c_{ij} = c_{ji}$, $I_{ij} = I_{ji}$. For any $t_i \in T$, $t_j \in T$ and arbitrary constants c_0 and I_0, if $c_{ij} = c_0$ and $I_{ij} = I_0$, then this version of the NTAP is called the uniform-cost NTAP (UCNTAP), otherwise it is called the nonuniform-cost NTAP (NUCNTAP). In addition, we assume that I_{ij} and c_{ij} are independent of computing nodes, which means that these computing nodes and communication network of multicore clusters to be considered in this paper are homogeneous.

4. Main Results

Some complexity problems of the NTAP on multicore clusters are analyzed in this section and the main analysis results of this paper are stated in Sections 4.1, 4.2, and 4.3.

4.1. Analysis of Communication Cost for a Single Computing Node

Theorem 1. *For the UCNTAP and any p_q with x_q tasks, if one supposes that every pair of x_q tasks communicates, then the total internode communication cost and the total intranode communication cost incurred on p_q are $x_q(n - x_q)c_0$ and $0.5x_q(x_q - 1)I_0$, respectively.*

Proof. If x_q tasks are allocated to p_q, then other $n - x_q$ tasks must be assigned to other computing nodes and there are $x_q(n - x_q)$ communications on p_q in all, and thus the total communication cost on p_q is equal to $x_q(n - x_q)c_0$. The intranode communication cost is only incurred between any two of the x_q tasks, and therefore the total intranode communication cost is $0.5x_q(x_q - 1)I_0$. □

Corollary 2. *For the NUCNTAP, the internode communication cost and the intranode communication cost incurred on p_q between any two tasks t_i and t_j are $x_{ijq}(2 - x_{ijq})c_{ij}$ and $0.5x_{ijq}(x_{ijq} - 1)I_{ij}$, respectively, where $x_{ijq} = x_{iq} + x_{jq}$.*

Proof. From Theorem 1, the total internode communication cost incurred on p_q equals $x_q(n - x_q)c_0$. When considering only two tasks t_i and t_j, $n = 2$, $c_0 = c_{ij}$ and $x_q = x_{iq} + x_{jq} = x_{ijq}$, the internode communication cost incurred on p_q is $x_{ijq}(2 - x_{ijq})c_{ij}$. Similarly, the intranode communication cost on p_q is $0.5x_{ijq}(x_{ijq} - 1)I_{ij}$. □

4.2. Complexity Analysis of the UCNTAP

Theorem 3. *The UCNTAP is a P-problem and can be solved in polynomial time if $I_0 \geq c_0$.*

Proof. (1) Transforming the UCNTAP into a minimum cost flow problem. As shown in Figure 1, the UCNTAP can be modeled as a minimum cost flow (MCF) problem on a network G. The ith task corresponds to a task vertex vt_i and all tasks correspond to a set $VT = \{vt_1, vt_2, \ldots, vt_n\}$. Similarly, the qth computing node p_q corresponds to a computing vertex vp_q and all computing nodes correspond to a set $VP = \{vp_1, vp_2, \ldots, vp_m\}$. The source r is connected to all task vertices by source edges of capacity 1 and cost 0, and all computing vertices are connected to the terminal s by terminal edges of capacity n and cost $0.5x_q(n - x_q)c_0 + 0.5x_q(x_q - 1)I_0$, where $1 \leq q \leq m$. Moreover, each task vertex is connected to all computing vertices by edges of capacity 1 and cost $e_{iq}x_{iq}$. In addition, we specify the initial amount of flow as n and the flows on all edges as integer flows.

(2) Proving the equivalence between the UCNTAP and the MCF problem, firstly, we prove that each feasible flow corresponds to a task assignment. With the initial amount of flow being n, for any $vt_i \in VT$, the amount of flow entering vt_i equals 1. According to flow conservation law, the amount of flow leaving vt_i is also equal to 1. As the flows on all edges are integer flows, the edges emanating from vt_i have one and only one edge of amount of flow 1. In other words, the ith task corresponding to vt_i is assigned to one and only one computing node p_{z_i}. Given any feasible flow F^0, without loss of generality, we assume that the set of all edges having amount of flow 1 and pointing to vertices of VP from vertices of VT is $\{(vt_1, vp_{z_1}), (vt_2, vp_{z_2}), \ldots, (vt_n, vp_{z_n})\}$; then, the feasible flow F^0 corresponds to a task assignment $Z^0 = (z_1, z_2, \ldots, z_n)$. Secondly, we prove that each task assignment corresponds to a feasible flow. Given any task assignment Z^0, we can construct a feasible flow F^0 in this way as follows. With the number of tasks being n, the initial amount of flow is n; that is, the amount of flow entering any $vt_i \in VT$ equals

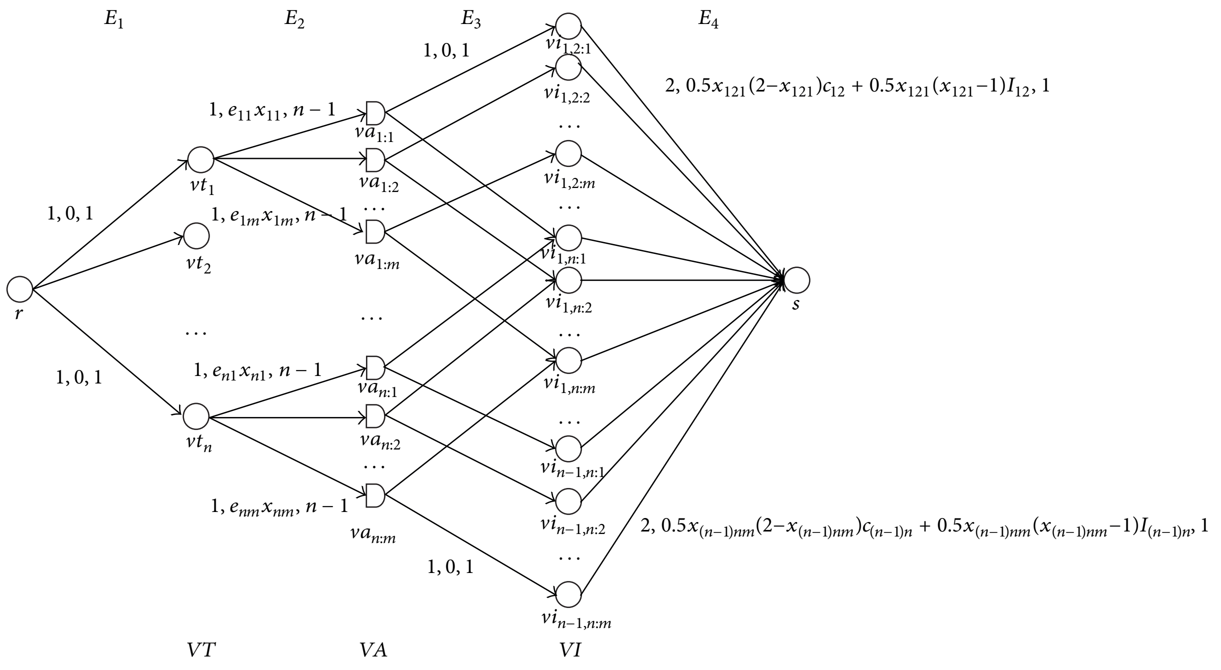


Figure 2: The generalized MCF problem equivalent to NUCNTAP.

1. If the ith task t_i is allocated to p_{z_i}, then the amount of flow on edge (vt_i, vp_{z_i}) equals 1. Therefore, we can construct a feasible flow F^0, on which all the edges having amount of flow 1 and pointing to computing vertices from task vertices constitute an edge set $\{(vt_1, vp_{z_1}), (vt_2, vp_{z_2}), \ldots, (vt_n, vp_{z_n})\}$. Lastly, we prove that the total cost of the feasible flow equals the total cost of corresponding task assignment and the MCF corresponds to an optimal task assignment. Clearly, the cost function $0.5x_q(n-x_q)c_0+0.5x_q(x_q-1)I_0$ of the MCF problem corresponds to the sum of internode communication cost and intranode communication cost, and $e_{iq}x_{iq}$ corresponds to execution cost. Hence, the total cost of any feasible flow equals the total cost of corresponding task assignment. In addition, for any MCF, we assume that F^* corresponds to a nonoptimal task assignment Z^0; that is, $C(F^*) = C(Z^0)$; then, there must exist an optimal task assignment Z^* such that $C(Z^*) < C(Z^0)$. Furthermore, the Z^* must correspond to a feasible flow F^0 such that $C(Z^*) = C(F^0)$, so $C(F^0) < C(F^*)$, which contradicts that F^* is a MCF. Thus, each MCF must correspond to an optimal task assignment.

(3) Analyzing the effect of communication cost on problem complexity. Now we analyze the effect of communication cost on the complexity of the NTAP by analyzing the effect of cost function on the complexity of the MCF problem. According to the construction process, the quadratic cost function of the MCF problem is given as

$$\begin{aligned} &0.5x_i(n-x_i)c_0 + 0.5x_i(x_i-1)I_0 \\ &\quad = 0.5(I_0 - c_0)x_i^2 + 0.5(nc_0 - I_0)x_i. \end{aligned} \tag{1}$$

The convexity/concavity of the quadratic cost function is determined by the quadratic coefficient $I_0 - c_0$. According to the positive/negative sign of $I_0 - c_0$, the MCF problem can be distinguished as

$$\begin{aligned} I_0 &= c_0, \quad \text{linear cost network MCF problem;} \\ I_0 &> c_0, \quad \text{convex cost network MCF problem;} \\ I_0 &< c_0, \quad \text{concave cost network MCF problem.} \end{aligned} \tag{2}$$

Here, the MCF problem is a P-problem in the cases of linear cost network and convex cost network, and the concave cost network MCF problem is a NP-hard problem. Hence, we can conclude that the UCNTAP is a P-problem if the intranode communication cost is not less than the internode communication cost and can be transformed into a convex cost network MCF problem. The convex cost network MCF problem can be solved in $O(M \log N(M + N \log N))$ time [16], where M denotes the number of edges and N denotes the number of vertices. Thus, the UCNTAP can be solved in $O(m^2n^2 \log(m+n))$ time if the intranode communication cost is not less than the internode communication cost, where m denotes the number of computing nodes or multicore nodes and n denotes the number of tasks. □

4.3. Complexity Analysis of the NUCNTAP

Theorem 4. *For any $t_i \in T$ and $t_j \in T$, the NUCNTAP is a P-problem and can be solved in polynomial time if the intranode communication cost is not less than the internode communication cost.*

Proof. (1) Transforming the NUCNTAP into a generalized network MCF problem. As shown in Figure 2, the NUCNTAP can be modeled as a generalized MCF problem on a network G, of which all vertices, with the exception of source vertex r and terminal vertex s, are divided into three levels. The first level is a task vertex level $VT = \{vt_1, vt_2, \ldots, vt_n\}$, where the vertex vt_i corresponds to the ith task t_i. The second level is a task assignment vertex level $VA = \{va_{1:1}, va_{1:2}, \ldots, va_{n:m}\}$. If the amount of flow through vertex $va_{i:q}$ equals 1 (0), then it denotes that t_i is (not) allocated to the qth computing node p_q. The third level is a task pair assignment vertex level $VI = \{vi_{1,2:1}, vi_{1,2:2}, \ldots, vi_{n-1,n:m}\}$. For the amount of flow through vertex $vi_{i,j:q}$, in case 0, it denotes that the ith task t_i and the jth task t_j are not assigned to p_q; in case 1, it denotes that t_i or t_j is allocated to p_q; in case 2, it denotes that t_i and t_j are both allocated to p_q.

The edges of network G can be divided into four levels. The first level is $E_1 = \{(r, vt_i)\}$, a set of edges having capacity 1, cost 0, and gain 1. The second level is $E_2 = \{(vt_i, va_{i:q})\}$, a set of edges having capacity 1, cost $e_{iq}x_{iq}$, and gain $n-1$. The third level is $E_3 = \{(va_{i:q}, vi_{i,j:q})\} \cup \{(va_{j:q}, vi_{i,j:q})\}$, a set of edges having capacity 1, cost 0, and gain 1. The fourth level is $E_4 = \{(vi_{i,j:q}, s)\}$, a set of edges having capacity 2, cost $0.5x_{ijq}(2 - x_{ijq})c_{ij} + 0.5x_{ijq}(x_{ijq} - 1)I_{ij}$, and gain 1. The cost network is a generalized cost network because the gain coefficients on edges of G are not all 1. In addition, we specify the initial amount of flow as n and the flows on all edges as integer flows.

(2) Proving the equivalence between the NUCNTAP and the generalized MCF problem. Firstly, we prove that each feasible flow corresponds to a task assignment. With the initial amount of flow being n, for any $vt_i \in VT$, the amount of flow entering vt_i is equal to 1. According to flow conservation law, for m edges leaving vt_i, there is one and only one edge of amount of flow 1 and all other $m-1$ edges have amount of flow 0. That is to say, the t_i corresponding to vt_i is allocated to one and only one computing node p_{z_i}. Given any feasible flow F^0, without loss of generality, we assume that the set of all edges having amount of flow 1 and pointing to vertices of VA from vertices of VT is $\{(vt_1, vp_{z_1}), (vt_2, vp_{z_2}), \ldots, (vt_n, vp_{z_n})\}$; then, the feasible flow F^0 corresponds to a task assignment $Z^0 = (z_1, z_2, \ldots, z_n)$. Secondly, we prove that each task assignment corresponds to a feasible flow. Given any task assignment $Z^0 = (z_1, z_2, \ldots, z_n)$, we can construct a feasible flow F^0 in this way as follows. With the number of tasks being n, the initial amount of flow is n; that is, the amount of flow entering any $vt_i \in VT$ is equal to 1. If t_i is allocated to p_{z_i}, then the amount of flow on edge $(vt_i, va_{i:z_i}) \in E_2$ is equal to 1. The amount of flow on E_3 can be determined after having determined the amount of flow on E_2. For any edge $(vt_i, va_{i:z_i}) \in E_2$ having amount of flow 1 and gain coefficient $n-1$, we can make the amount of flow leaving $va_{i:z_i}$ to be $n-1$. Thereby, the amount of flow on each of $n-1$ edges leaving $va_{i:z_i}$ and having capacity 1 equals 1 and we can construct a feasible flow F^0, where the edges of amount of flow 1 of E_2 constitute an edge set $\{(vt_1, va_{1:z_1}), (vt_2, va_{2:z_2}), \ldots, (vt_n, va_{n:z_n})\}$. Lastly, we prove that the total cost of the feasible flow equals the total cost of corresponding task assignment and the MCF corresponds to an optimal task assignment. Clearly, the cost function $0.5x_{ijq}(2 - x_{ijq})c_{ij} + 0.5x_{ijq}(x_{ijq} - 1)I_{ij}$ of the generalized network MCF problem corresponds to the sum of internode communication cost and intranode communication cost, and $e_{iq}x_{iq}$ corresponds to execution cost. Therefore, the total cost of any feasible flow equals the total cost of corresponding task assignment. For any MCF F^*, we assume that F^* corresponds to a non-optimal task assignment Z^0; namely, $C(F^*) = C(Z^0)$; then, there must exist an optimal task assignment Z^* such that $C(Z^*) < C(Z^0)$. Furthermore, the Z^* must correspond to a feasible flow F^0 such that $C(Z^*) = C(F^0)$, so $C(F^0) < C(F^*)$, which contradicts that the F^* is a MCF. Thus, each MCF must correspond to an optimal task assignment.

(3) Analyzing the effect of communication cost on problem complexity, we analyze the effect of internode communication cost and intranode communication cost on the complexity of the NTAP by analyzing the effect of cost function on the complexity of the generalized network MCF problem. According to the construction process, the quadratic cost function of the generalized network MCF problem is given as

$$\begin{aligned} &0.5x_{ijq}\left(2 - x_{ijq}\right)c_{ij} + 0.5x_{ijq}\left(x_{ijq} - 1\right)I_{ij} \\ &\quad = 0.5\left(I_{ij} - c_{ij}\right)x_{ijq}^2 + \left(c_{ij} - 0.5I_{ij}\right)x_{ijq}. \end{aligned} \tag{3}$$

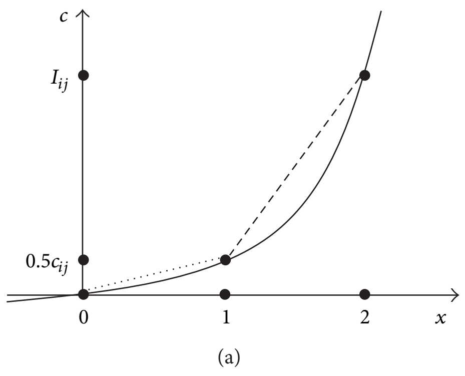
The convexity/concavity of the quadratic cost function is determined by the quadratic coefficient $I_{ij} - c_{ij}$. According to the positive-negative sign of $I_{ij} - c_{ij}$, the MCF problem can be distinguished as

$$\begin{aligned} I_{ij} = c_{ij}, \quad & \text{generalized linear cost} \\ & \text{network MCF problem;} \\ I_{ij} > c_{ij}, \quad & \text{generalized convex} \\ & \text{cost network MCF problem;} \\ I_{ij} < c_{ij}, \quad & \text{generalized concave} \\ & \text{cost network MCF problem.} \end{aligned} \tag{4}$$

Here, the generalized linear cost network MCF problem and generalized convex cost network MCF problem are P-problem, and the generalized concave cost network MCF problem is a NP-hard problem. Hence, we can conclude that the NUCNTAP is a P-problem if the intranode communication cost is not less than the internode communication cost and can be cast as a convex cost network MCF problem. The generalized convex cost network MCF problem can be solved in $O(MN)$ time [16], where M denotes the number of edges and N denotes the number of vertices. Thus, the NUCNTAP can be solved in $O(m^2n^4)$ time if the intranode communication cost equals the internode communication cost, where m denotes the number of computing nodes or multicore nodes and n denotes the number of tasks. □

5. Discussing Solutions to the NTAP

The effects of communication cost on complexity of the NTAP have been analyzed and proven. Further, solutions



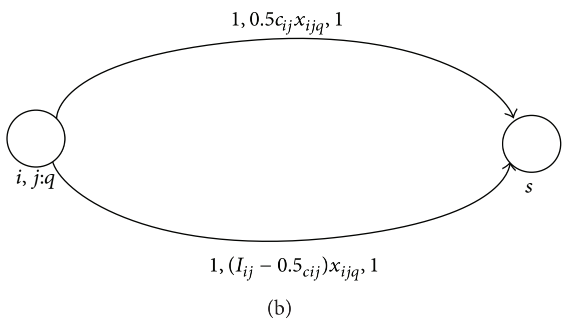


Figure 3: The piecewise linear approximation representation for convex cost function: (a) convex cost function and (b) piecewise linear approximation representation.

to the NTAP are discussed in this section. Unfortunately, Bokhari [9] has shown that the traditional TAP for four or more processors is a NP-complete problem. Needless to say, the NTAP can be difficult. Therefore, solving the NTAP is a challenging problem.

The NTAP can be modeled as a generalized network flow model and thus can be solved with minimum cost flow algorithms. However, solutions should have much difference in complexity due to the convexity/concavity of minimum cost flow problems [17]. In general, the NTAP is a NP-hard problem and cannot be solved in polynomial time, which usually is solved with approximation algorithms or heuristic suboptimal algorithms [5]. When the intranode communication cost equals the internode communication cost, the NTAP can be cast as a linear network minimum cost flow problem and can be solved with flow augmentation approach or primal approach [17]. When the intranode communication cost is more than the internode communication cost, the convex network minimum cost flow can be converted into a linear network minimum cost flow and thus can be solved with flow augmentation method or primal approach. The transformation process is shown in Figure 3. Convex cost on edge of set E_4 in Figure 2 can be approximately represented as piecewise linear cost and each convex cost curve shown in Figure 3(a) can be approximately represented as two linear cost edges or arcs shown in Figure 3(b). Thus, the convex network minimum cost flow problem can be converted into a linear network minimum cost flow problem to be solved.

Furthermore, the mathematical programming model corresponding to the model represented in Figure 2 can be modeled as formulation (5). Thus, the NTAP also can be solved with mathematical programming approaches, where $E^+(i)$ denotes the outgoing edge set of vertex i and $E^-(i)$ denotes the incoming edge set of vertex i, $q_k = e_{ij}$, k denotes edge $(vt_i, va_{i:j})$; $u_k = [(1 - 0.5x_{ijq})c_{ij} + 0.5(x_{ijq} - 1)I_{ij}]x_{ijq}$, k denotes edge $(vi_{i,j:q}, s)$, f_k denotes amount of flow on edge k, and V denotes vertex set of G.

In fact, the excellent results, as shown in [18], demonstrate that solution to the NTAP presented in this paper is particularly efficient when a large number of tasks communicate, solving reasonably large problems faster than other exact approaches available:

$$\begin{aligned}
\text{Min} \quad & \sum_{k\in E_2} q_k x_k + \sum_{k\in E_4} u_k x_k, \\
\text{s.t.} \quad & -\sum_{k\in E^-(i)} f_k + \sum_{k\in E^+(i)} f_k = 0, \quad i \in V \setminus \{r, s, VA\}, \\
& -\sum_{k\in E^-(i)} (n-1) f_k + \sum_{k\in E^+(i)} f_k = 0, \quad i \in VA, \\
& -\sum_{k\in E^-(r)} f_k + \sum_{k\in E^+(r)} f_k = n, \\
& -\sum_{k\in E^-(s)} f_k + \sum_{k\in E^+(s)} f_k = -n(n-1), \\
& 0 \le f_k \le 2, \quad k \in E_4, \\
& 0 \le f_k \le 1, \quad k \in E_1 \cup E_2 \cup E_3.
\end{aligned} \tag{5}$$

6. Conclusions

This paper investigates the effects of communication cost on complexity of the NTAP and demonstrates the relationships between complexity and communication cost. We also have proved that (1) the NTAP can be solved in $O(m^2n^4)$ time if the intranode communication cost equals the internode communication cost; (2) the NTAP can be solved in polynomial time if the intranode communication cost is more than the internode communication cost and specifically, the UCNTAP can be solved in $O(m^2n^2\log(m+n))$ time; (3) the NTAP is a NP-hard problem if the intranode communication cost is less than the internode communication cost, which indicates that efficient polynomial time algorithms still remain to be further investigated. Furthermore, solutions to the NTAP are also discussed and need to be further studied. Our work extends currently known theoretical results and the theorems and conclusions presented in this paper can provide theoretical basis for task allocating strategies in multicore cluster systems.

Conflict of Interests

The author declares no competing financial interests.

Acknowledgments

This work is partially supported through the support from NSFC (60873256), the National Basic Research 973 Program of China (2005CB321904), and the Qinhuangdao Science and Technology Support Program (201302A038).

References

[1] L. Chai, Q. Gao, and D. K. Panda, "Understanding the impact of multi-core architecture in cluster computing: a case study with Intel dual-core system," in *Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07)*, pp. 471–478, Rio de Janeiro, Brazil, May 2007.

[2] X. F. Wu, V. Taylor, C. Lively, and S. Sharkawi, "Performance analysis and optimization of parallel scientific applications on CMP clusters," *Scalable Computing*, vol. 10, no. 1, pp. 61–74, 2009.

[3] J. Parrow, "An introduction to the π-calculus," in *Handbook of Process Algebra*, J. A. Bergstra, A. Ponse, and S. A. Smolka, Eds., pp. 479–543, Elsevier Science, Amsterdam, The Netherlands, 2001.

[4] K. Asanovic, R. Bodik, B. C. Catanzaro et al., "The landscape of parallel computing research: a view from Berkeley," Tech. Rep. UCB/EECS-2006-183, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Calif, USA, 2006.

[5] A. Ernst, H. Jiang, and M. Krishnamoorthy, "Exact solutions to task allocation problems," *Management Science*, vol. 52, no. 10, pp. 1634–1646, 2006.

[6] H. S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," *IEEE Transactions on Software Engineering*, vol. 3, no. 1, pp. 85–93, 1977.

[7] S. H. Bokhari, "Dual processor scheduling with dynamic reassignment," *IEEE Transactions on Software Engineering*, vol. 5, no. 4, pp. 341–349, 1979.

[8] H. S. Stone, "Program assignment in three-processor systems and tricutset partitioning of graphs," Tech. Rep. ECE-CS-77-7, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, Mass, USA, 1977.

[9] S. H. Bokhari, "Shortest tree algorithm for optimal assignments across space and time in a distributed processor system," *IEEE Transactions on Software Engineering*, vol. 7, no. 6, pp. 583–589, 1981.

[10] D. Towsley, "Allocating programs containing branches and loops within a multiple processor system," *IEEE Transactions on Software Engineering*, vol. 12, no. 10, pp. 1018–1024, 1986.

[11] D. Fernandez-Baca, "Allocating modules to processors in a distributed system," *IEEE Transactions on Software Engineering*, vol. 15, no. 11, pp. 1427–1436, 1989.

[12] C.-H. Lee, D. Lee, and M. Kim, "Optimal task assignment in linear array networks," *IEEE Transactions on Computers*, vol. 41, no. 7, pp. 877–880, 1992.

[13] S. Y. Cho and K. H. Park, "Dynamic task assignment in heterogeneous linear array networks for metacomputing," in *Proceedings of the Heterogeneous Computing Workshop*, pp. 66–71, Cancún, Mexico, April 1994.

[14] W. Fernandez de la Vega and M. Lamari, "The task allocation problem with constant communication," *Discrete Applied Mathematics*, vol. 131, no. 1, pp. 169–177, 2003.

[15] P. K. Yadav, M. P. Singh, and H. Kumar, "Scheduling algorithm: tasks scheduling algorithm for multiple processors with dynamic reassignment," *Journal of Computer Systems, Networks, and Communications*, vol. 2008, Article ID 578180, 9 pages, 2008.

[16] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, NJ, USA, 1993.

[17] P. A. Jensen and W. J. P. Barnes, *Network Flow Programming*, Krieger Publishing Company, Malabar, Fla, USA, 1987.

[18] J. X. Yang, G. Z. Tan, F. Wang, and D. Pan, "Solution to new task allocation problem on multi-core clusters," *Journal of Computational Information Systems*, vol. 7, no. 5, pp. 1691–1697, 2011.