Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 708495, 16 pages
http://dx.doi.org/10.1155/2013/708495

Research Article

A Genetic Algorithm for Task Scheduling on NoC Using FDH Cross Efficiency

Song Chai, Yubai Li, Jian Wang, and Chang Wu

School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Song Chai; s.tschai@gmail.com

Received 5 August 2013; Accepted 7 November 2013

Academic Editor: Hao-Chun Lu

A CrosFDH-GA algorithm is proposed for the task scheduling problem on the NoC-based MPSoC regarding the multicriterion optimization. First of all, four common criterions, namely, makespan, data routing energy, average link load, and workload balance, are extracted from the task scheduling problem on NoC and are used to construct the DEA DMU model. Then the FDH analysis is applied to the problem, and a FDH cross efficiency formulation is derived for evaluating the relative advantage among schedule solutions. Finally, we introduce the DEA approach to the genetic algorithm and propose a CrosFDH-GA scheduling algorithm to find the most efficient schedule solution for a given scheduling problem. The simulation results show that our FDH cross efficiency formulation effectively evaluates the performance of schedule solutions. By conducting comparative simulations, our CrosFDH-GA proposal produces more metrics-balanced schedule solution than other multicriterion algorithms.

1. Introduction

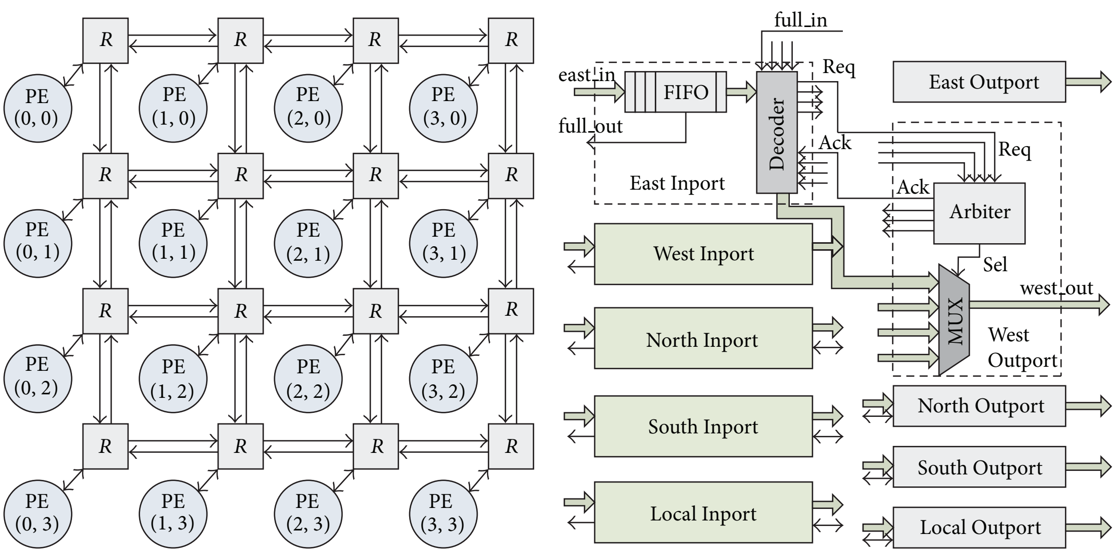
The scheduling problem has long been a research hotspot since its proposal in the 1950s, as the Job-shop scheduling problem [1]. After the computer technology emerged in the 1940s, the scheduling problem also found its position in the computer science region, as the task scheduling problem on the uniprocessor in the 1960s [2], the multiprocessor in the 1970s [3], the distributed computing in the 1980s [4], and the grid computing in the early 21st century [5]. Now the chip fabrication technology has brought us to the single-chip multicore era [6]. The presence of chip multiprocessor (CMP), especially the NoC (network-on-chip)-based MPSoC [7], brings new challenges to the task scheduling algorithm design.

In NoC solution, the idea of introducing the network infrastructure to the chip design, along with the newly arisen concept of green communication [8], makes the goal of scheduling algorithm change from single-objective optimization on makespan to performing optimization simultaneously on multiple metrics, not only the traditional makespan, but also energy [9] and NoC criterions [10], and some of the optimizations of these metrics are even in conflict with each other. So the goal of scheduling algorithm design leans forward to balancing these multiple metrics.

On the other hand, DEA is a nonparametric technique that is used to measure the relative efficiency of multi-input multioutput DMUs (decision making units). It was first presented by Charnes, Cooper, and Rhodes as the CCR model in 1976 [11], then developed into several variations based on different RTS (return-to-scale) assumptions. The concept of efficiency in DEA gives us a reasonable standard to make trade-off between multiple metrics.

In this paper, a FDH DEA model of NoC task scheduling is constructed, and a FDH cross efficiency formulation is proposed based on peer appraisal for further assessment of the relative advantages of DMUs. Then the proposed DEA approach is introduced to the genetic algorithm, and a CrosFDH-GA scheduling algorithm is proposed for the task scheduling problem on NoC to find the most efficient and balanced schedule solution.

The rest of this paper is organized as follows. Section 2 summarizes the related work of this paper; Section 3 formulates the task scheduling problem on NoC; the FDH and cross efficiency FDH formulation is given in Section 4; Section 5 presents our CrosFDH-GA scheduling algorithms;



(a) A 4 × 4 2D mesh NoC

(b) Microstructure of NoC node

Figure 1: Architecture of NoC-based MPSoC.

simulations results and discussion are given in Section 6; Section 7 concludes the paper.

2. Related Works

The multicriterion scheduling algorithms for CMPs have been widely researched in recent years. In [12], a scheduling algorithm is proposed for multicore processors to avoid resource contention, as well as to reduce energy consumption. A modified genetic algorithm which incorporates bacteriological algorithm is proposed in [13] to maximize the system reliability and reduce makespan. In [14], the optimization of makespan and workload balance is addressed, and a NSGA-II based schedule algorithm is proposed for multicore-based grid. A multiobjective evolutionary algorithm (MOEA) based schedule heuristic is proposed for the joint optimization of performance, energy, and temperature on multicore processors in [15].

As for the DEA's application in the field of task scheduling on multicore, a FDH-based evaluation method for the assessment of schedule heuristics is proposed in [16]. Although both [16] and our work adopt DEA FDH model as the analytic tool, our work introduces the concept of cross efficiency to the FDH model and uses FDH cross efficiency to rank schedules. Moreover, the incorporation of DEA evaluation method into metaheuristic also distinguishes itself from the work in [16].

3. Problem Formulation

3.1. Task Model. In this paper, tasks are modeled using directed acyclic graphs (DAGs). A DAG $G = (V, E)$ is an acyclic graph where V is the set of nodes which represent the tasks and E is the set of edges in which an element e_{ij} denotes the communication from task i to task j. The edge indicates the precedent relation between two tasks.

Each node v_i and edge e_{ij} are associated with a weight, denoted by C_i and T_{ij}, respectively. Weight C_i is the computational load required by a processing Element (PE) to execute task i; and T_{ij} is the data transmission load between task i and task j. In our work, both C_i and T_{ij} are presented using time unit (cycles).

3.2. Network-on-Chip Hardware. The target hardware is a 2D mesh NoC-based MPSoC, as illustrated in Figure 1(a). Each PE is connected to a router, and routers are interconnected with each other through bidirection links. Data is transferred through NoC in the form of packets.

PEs are homogenous processor cores with local data cache. If two consequential tasks are scheduled to the same PE, the successor task reads the predecessor's data directly from the data cache of the PE without routing in NoC.

The microstructure of a NoC router is shown in Figure 1(b). The router has five *Inports* and *Outports* corresponding to five directions of *East*, *West*, *North*, *South*, and *Local*. The decoder in the *Inport* scans the first flit of the FIFO for any incoming packet. If the decoder detects the head flit of a packet, it performs *XY routing algorithm* and sends request signal to the arbiter of the corresponding *Outport*. If the arbiter receives multiple request signals, the contention is solved using *Round-Robin arbitration*. The granted *Inport* then forwards the packet to the downstream router. *Wormhole routing* is adopted to minimize the buffer requirement as well as the packet latency [17]. The *back pressure* mechanism is also employed to further reduce end-to-end delay [18].

The energy model of a NoC is presented by the *Bit Energy* proposed in [19]. Analytically, the average energy consumption of transmitting one bit from node i to node j is calculated by

$$E_{\text{bit}}^{i,j} = n_{\text{hops}} \cdot E_{N_{\text{bit}}} + \left(n_{\text{hops}} - 1\right) \cdot E_{L_{\text{bit}}}, \quad (1)$$

where $E_{N_{\text{bit}}}$ and $E_{L_{\text{bit}}}$ represent the energy consumed on the node and on the link, respectively, and n_{hops} is the number of nodes the bit passes on its way from node i to node j.

3.3. Monitored Metrics. In this paper, four common metrics of NoC are extracted and monitored for the assessment of schedule solution, and they are makespan, data routing energy, average link load [20], and workload balance [21].

Makespan (the M metric), which is the amount of time required by a NoC to finish entire tasks in a DAG following the instruction of a schedule, is the time metric of a schedule, while the *data routing Energy* (the E metric) is the total amount of energy dissipated in each NoC component during the execution of the DAG.

The *average Link load* (the L metric) is calculated by adding up all the data transmission time (in cycles) on each link and then dividing it by the number of links (for the NoC in Figure 1(a), there are 48 links). The L metric represents how busy the NoC is during the execution, and a higher average link load metric implies a higher possibility of link contentions. A good schedule is supposed to reduce the L metric.

Finally, the *workload Balance* (the B metric) is defined to be the inverse coefficient of variant of the total workload on each processor, as shown in (2). The $\text{load}_{\text{prc}}(n)$ is the actual load on processor n, and the load_{ave} is the average load. The B metric reflects the load balance of the processors

$$\text{workload balance} = \frac{\text{load}_{\text{ave}}}{\left(\sum_n \left(\text{load}_{\text{ave}} - \text{load}_{\text{prc}}(n)\right)^2\right)^{1/2}}. \tag{2}$$

4. DEA Evaluation of Schedules

4.1. A Brief Review of DEA. Data envelopment analysis (DEA) is a nonparametric technique that is widely used to measure the relative efficiency among many-input, many-output decision-making units (DMUs), which in our context are the schedules. The efficiency of a DMU is defined as the weighted sum of its output divided by the weighted sum of its input. The essence of DEA is that it allows each DMU to choose a particular set of weight coefficients which favors its own efficiency, under the constraint that the efficiencies of all DMUs calculated by this set of coefficients do not exceed 1. A DMU is "efficient" if the efficiency calculated by DEA is 1; otherwise the DMU is marked as "inefficient." The following *Linear Programing* (LP) problem is the CCR model of DEA: (CCR multiplier form)

$$\begin{aligned} \max \quad & Z = u^T \cdot y_i \\ \text{s.t} \quad & \begin{cases} v \cdot x_i = 1, \\ v \cdot x_l - u \cdot y_l \geq 0, \quad l = 1, 2, \ldots, n, \\ v, u \geq 0, \end{cases} \end{aligned} \tag{3}$$

where $x_i = (x_{i,1} \quad x_{i,2} \quad \cdots \quad x_{i,m})^T \in R^m$ and $y_i = (y_{i,1} \quad y_{i,2} \quad \cdots \quad y_{i,s})^T \in R^s$ are the input and output vectors of DMU i; $v = (v_1 \quad v_2 \quad \cdots \quad v_m)$ and $u = (u_1 \quad u_2 \quad \cdots \quad u_s)$ are the coefficient (multiplier) vectors of the inputs and outputs; n is the number of DMUs; and the objective function Z is the efficiency of DMU i.

The result of applying DEA is a classification among DMUs as efficient group or inefficient group. The efficient DMUs form an efficient frontier on the multi-input, multioutput space that envelops all inefficient DMUs. The projection of inefficient DMU on the efficient frontier is the hypothetical efficient unit, which is a linear combination of the efficient DMUs. An inefficient DMU can also be converted to an efficient DMU by proportionally scaling down by the value of its efficiency in the inputs and maintaining its original outputs. This interpretation of DEA efficiency is corresponding to the envelop form of DEA, which is the dual problem of (3) (CCR envelop form)

$$\begin{aligned} \min_{\theta,\lambda} \quad & Z = \theta \\ \text{s.t} \quad & \begin{cases} \theta x_i - X \cdot \lambda \geq \mathbf{0}, \\ y_i - Y \cdot \lambda \leq \mathbf{0}, \\ \lambda \geq 0, \end{cases} \end{aligned} \tag{4}$$

where $X = (x_1 \quad x_2 \quad \cdots \quad x_n)$ and $Y = (y_1 \quad y_2 \quad \cdots \quad y_n)$ are the input and output matrices; $\lambda = (\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_n)^T \in R^n$ is a nonnegative vector; and θ is the efficiency of DMU i.

The way that the efficient frontier is generated differentiates between DEA models which imply different returns to scale assumptions. There are four basic returns to scale assumption: constant returns to scale (CRS), corresponding to the CCR model [11]; variable returns to scale (VRS), corresponding to the BCC model [22]; increasing returns to scale; and decreasing returns to scale.

In this paper, we focus on a special case of DEA, namely, the free disposal hull (FDH) [23]. In VRS FDH formulation of DEA, each DMU is evaluated by comparing itself to other DMUs on a one-on-one basis, and a DMU is considered efficient only when no other DMU dominates it.

In most cases, the idea of DEA that let DMU specify its own weight coefficients to show its maximum advantage is desirable. However, in some extreme scenes, a DMU can "cheat" a high efficiency score by weighting a single input or a single output and setting the rest weight coefficients close to 0. This can happen when some DMUs have a particularly small input or particularly large output; in our words, these DMUs have unbalanced metrics, and these "mavericks" need to be depreciated. Moreover, although DEA effectively discriminates between efficient and inefficient DMUs, it does not further assess the relative advantages among the efficient ones.

One solution to the above questions is to introduce cross efficiency in the efficiency measurement. The concept of cross efficiency, corresponding to the simple efficiency implied by original DEA, is the peer-appraisal equivalent of DEA's self-appraisal process. The cross efficiency of a certain DMU is the efficiency value calculated by using weight coefficients derived by other DMUs. If a DMU is a maverick or has unbalanced metrics, then its cross efficiency value derived from other DMU's coefficients is not likely to be high.

Two widely accepted crossefficiency formulations, the aggressive and benevolent formulations, were proposed in [24] based on CCR Model. Both formulations add a secondary goal to the normal DEA efficiency calculation (maximizing reference DMU's efficiency): the aggressive formulation minimizes target DMU's crossefficiency, while the benevolent formulation maximizes its efficiency. Given that the simple DEA efficiency of DMU i is E_{ii}, then the crossefficiency of DMU j evaluated by DMU i is defined by (CCR crossefficiency, benevolent formulation)

$$\begin{aligned} \max \quad & E_{ji} = u^T \cdot y_j \\ \text{s.t} \quad & \begin{cases} v \cdot x_j = 1, \\ v \cdot x_l - u \cdot y_l \geq 0, \quad l = 1, 2, \ldots, n, \\ E_{ii} \cdot v \cdot x_i - u \cdot y_i = 0, \\ v, u \geq 0. \end{cases} \end{aligned} \tag{5}$$

4.2. FDH DEA Evaluation of Schedules. In order to apply data envelopment analysis to the schedule evaluation, the multi-input multioutput DMU model needs to be defined using the schedule metrics, namely, *makespan* (M), *routing energy* (E), *average link load* (L) and *workload balance* (B), proposed in Section 3.

The classification of metrics as inputs and outputs follows a simple "rule of thumb" [16]. If the value of a metric is larger-is-better, then it is an output; otherwise the metric is an input. As a result of this classification, the DMU model of our schedule evaluation is as follows: $x = (M \quad E \quad L)^T$ are the inputs, and $y = B$ is the output.

Moreover, in the rest of this paper the term "schedule" is referring to a schedule that is discriminable using the four-metric classification. If two schedules have identical metrics, they are regarded as the same schedule; at least they are not discriminable under current metrics.

With the scheduling DMU model, the observed schedules are defined as follows.

Definition 1 (schedule set). The observed schedules or the *schedule set*$\Gamma = \{\gamma_1, \ldots, \gamma_i, \ldots, \gamma_n\} = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$ is the set of n observed schedules, where each schedule γ_i is defined by the input vector $x_i = (M_i \quad E_i \quad L_i)^T$ and output scalar $y_i = B_i$.

Another concept that is relevant to DEA is the *possible production set*. The *possible production set* is the space enveloped by the efficient frontier on the multi-input multioutput space. Normally, the *possible production set* is unknown in the DEA and needs to be constructed using the observed DMUs. The FDH *possible production set* postulates were proposed in [23]. Here, under our context, we restate these postulates as the following axiom.

Axiom 1 (possible schedule set). The *possible schedule set* (PSS) of a schedule set Γ satisfies the following.

Postulate I. PSS contains all schedules in Γ.

Postulate II. PSS contains unobserved schedule γ_i if

(i) $\gamma_j = (x_j, y_j) \in \Gamma$, and $x_i = (M_i \quad E_i \quad L_i) \geq x_j$ or

(ii) $y_i = B_i \leq y_j$.

Postulate II is the free disposal postulate, which suggests a free disposal hull of Γ [25]. Together with the determinist Postulate I, they define the FDH *possible schedule set* of our schedule efficiency analysis. Now, we formally introduce FDH DEA to the schedule evaluation and define the efficient schedule as follows.

Definition 2 (efficient schedule and efficient schedule set). In a schedule set Γ, a schedule γ_i is called efficient if the optimization problem (6) has an optimal solution of $Z^* = 1$; otherwise γ_i is inefficient. The set of all efficient schedules in Γ is called the efficient schedule set, denoted by Γ_e, and likewise, the inefficient schedule set is denoted by Γ_i:

$$\begin{aligned} \min_{\theta, \lambda, s^+, s^-} \quad & Z = \theta - \varepsilon \cdot \left(\mathbf{1} \cdot s^+ + s^-\right) \\ \text{s.t} \quad & \begin{cases} \theta x_i - X \cdot \lambda - s^+ = \mathbf{0}, \\ y_i - Y \cdot \lambda + s^- = 0, \\ \mathbf{1} \cdot \lambda = 1, \lambda_l \in \{0, 1\}, \quad l = 1, 2, \ldots, n, \\ s^+, s^- \geq 0, \end{cases} \end{aligned} \tag{6}$$

where $x_i = (M_i \quad E_i \quad L_i)^T$ and $y_i = B_i$ are the input vector and output scalar of γ_i; n is the number of schedules in Γ; $X = (x_1 \quad x_2 \quad \cdots \quad x_n)$ and $Y = (y_1 \quad y_2 \quad \cdots \quad y_n)$ are the input and output matrices of Γ; $\lambda = (\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_n)^T$ is a binary vector in R^n; s^+ and s^- are the nonnegative slack variables, representing input excess and output shortfall; ε is a non-Archimedean infinitesimal constant; and θ is the efficiency of γ_i.

Optimization problem (6) is a *Mixed-Integer Programing* (MIP). The binary vector λ and the constraint $\mathbf{1} \cdot \lambda = 1$ enforce a one-on-one comparison between target schedule γ_i and all the schedules in Γ to search for a reference schedule γ_r that minimizes θ.

The first two constraints of problem (6) can be simplified to (7) by removing the slack variables. Obviously, the problem has feasible solutions when $l = i$ and $\theta = 1$ in this situation. From this point on, if a reference schedule γ_r is found with $\theta < 1$, that means that γ_r produces output y_r at least as same as γ_i, with the inputs no more than θx_i, which is a scale-down from x_i. Then this makes γ_i inefficient:

$$\begin{gathered} \theta x_i \geq x_l, \\ y_i \leq y_l, \quad l = 1, 2, \ldots, n. \end{gathered} \tag{7}$$

However, only $\theta = 1$ is not enough for a schedule to be efficient. Consider an efficient schedule $\gamma = (M \quad E \quad L \quad B)$ and an inefficient schedule $\widehat{\gamma} = (M + \Delta_m \quad E \quad L \quad B)$ which is distinguished from γ by a small input excess Δ_m in the makespan; the constraint (7) holding for γ also holds for $\widehat{\gamma}$; thus $\theta = 1$ for $\widehat{\gamma}$. The slacks variables s^+ and s^- are introduced to remove these schedules with input excess and output shortfall. The nonzero s^+ and s^- force the objective function in (6) to be less than 1.

Definition 1 implies the dominance/Pareto optimality of an efficient schedule. A dominant schedule is defined as follows.

Definition 3 (dominant schedule). A schedule γ_a is said to dominate γ_b if

(I) each component of $x_a = (M_a \quad E_a \quad L_a)^T$ is not greater than that of x_b;

(II) $y_a = B_a$ is not less than y_b;

(III.a) at least of component of x_a is less than the corresponding one of x_b (dominates in input); or

(III.b) y_a is greater than y_b (dominates in output).

Then the relationship between efficient schedule and dominant schedule is given in Theorem 4.

Theorem 4 (dominance and efficiency). *In a schedule set* Γ, *a schedule* γ_i *is called efficient if and only if no other schedule dominates it.*

Proof. If Part. Suppose there is a schedule γ_j that dominates γ_i.

The efficiency of γ_i is calculated by solving the following optimization problem:

$$\min_{\theta,\lambda,s^+,s^-} \quad Z = \theta - \varepsilon \cdot (s_M^+ + s_E^+ + s_L^+ + s_B^-)$$

$$\text{s.t} \quad \begin{cases} \theta \cdot M_i - (M_1 \quad M_2 \quad \cdots \quad M_n) \cdot \lambda - s_M^+ = 0, \\ \theta \cdot E_i - (E_1 \quad E_2 \quad \cdots \quad E_n) \cdot \lambda - s_E^+ = 0, \\ \theta \cdot L_i - (L_1 \quad L_2 \quad \cdots \quad L_n) \cdot \lambda - s_L^+ = 0, \\ B_i - (B_1 \quad B_2 \quad \cdots \quad B_n) \cdot \lambda + s_B^- = 0, \\ \sum \lambda_l = 1, \quad \lambda_l \in \{0,1\}, \quad l = 1,2,\ldots,n. \end{cases} \tag{8}$$

Let $\lambda_j = 1$ and $\theta = 1$; then constraints of (8) are converted to

$$\begin{aligned} M_i - M_j - s_M^+ &= 0, \\ E_i - E_j - s_E^+ &= 0, \\ L_i - L_j - s_L^+ &= 0, \\ B_i - B_j + s_B^- &= 0. \end{aligned} \tag{9}$$

From the domination relation, we have $M_i \geq M_j$, $E_i \geq E_j$, $L_i \geq L_j$, and $B_i \leq B_j$, and at least one of above inequations holds strictly. This means at least one of s_M^+, s_E^+, s_L^+, and s_B^- in (4) is not zero. So the objective function Z in (8) has a feasible solution of $Z' = 1 - \varepsilon \cdot (s_M^+ + s_E^+ + s_L^+ + s_B^-) < 1$, and schedule solution γ_i is not efficient.

Only If Part. Suppose schedule γ_i is not efficient then there must exist a set of θ^*, λ^*, s^{+*}, and s^{-*} that makes the optimal value of (8) be $Z^* < 1$. Assuming $\lambda_j = 1$ in λ^*, from (8) we have

$$\begin{aligned} \theta^* \cdot M_i &= M_j + s_M^{+*}, \\ \theta^* \cdot E_i &= E_j + s_E^{+*}, \\ \theta^* \cdot L_i &= L_j + s_L^{+*}, \\ B_i &= B_j - s_B^{-*}. \end{aligned} \tag{10}$$

The optimal value of $Z^* = \theta^* - \varepsilon \cdot (s_M^{+*} + s_E^{+*} + s_L^{+*} + s_B^{-*}) < 1$ suggests that (I) $\theta^* = 1$, and at least one of the s^{+*} and s^{-*} is not zero or (II) $\theta^* < 1$. Either of the above situations implies that schedule γ_i is dominated by γ_j. □

Theorem 4 relates the relatively abstract concept of FDH efficiency to the concept of dominance. Moreover, the following two corollaries are deduced from Theorem 4.

Corollary 5. *In a schedule set* Γ, *if a schedule* γ_i *satisfies*

(1) $M_i < M_l$, *or*

(2) $E_i < E_l$, *or*

(3) $L_i < L_l$, *or*

(4) $B_i > B_l$, $(l = 1, 2, \ldots, n, l \neq i)$,

then it is in the efficient schedule set Γ_e.

Proof. If γ_i satisfies one of the above conditions, then there is no schedule dominating γ_i. From Theorem 4, γ_i is efficient. □

Corollary 5 points out that the schedule with the smallest makespan or the smallest energy consumption or the smallest queuing time or the best workload balance is an efficient schedule.

Corollary 6. *In a schedule set* Γ, *removing any inefficient schedule from* Γ *does not change the elements in* Γ_e.

Proof. The schedules in Γ_e are dominant ones, and removing any inefficient schedule will not change the dominant position of the elements in Γ_e. Thus, Γ_e remains unchanged. □

4.3. FDH Cross Evaluation of Schedules. In this section, a cross evaluation process is proposed based on peer-appraisal FDH DEA for further assessment of schedules. DEA calculates the efficiency of a DMU by allowing the DMU to choose a scenario that is best for itself. Although this basis is plausible in most situations, some "maverick" DMUs, especially the DMUs with a single small input or a single large output, may "cheat" DEA to achieve high score by valuing its only strength and depreciating other metrics. These unbalanced DMUs must be devaluated during further assessment. Moreover, an assessment of relative advantages among efficient DMUs is also required.

FDH model is a MIP problem in nature. In order to derive its peer-appraisal variation, the dual problem of the FDH DEA in (6) is needed to construct the formulation. Normally, it is difficult to write the dual problem of a MIP; however, by exploring the particularity of the vector λ, Agrell has proven in [26] that the MIP problem in FDH model can be simplified to a LP problem. Using his result, the FDH envelop form in (6) is reduced to the LP problem given in

$$\min_{\theta_l,\lambda,s^+,s^-} \quad Z = \sum_l \theta_l + \varepsilon \cdot \sum_l \left(\mathbf{1} \cdot s_l^+ + s_l^-\right)$$
$$\text{s.t} \quad \begin{cases} \theta_l x_i - x_l \cdot \lambda_l - s_l^+ = 0, & l = 0,1,\ldots,n, \\ \left(y_i - y_l\right) \cdot \lambda_l + s_l^- = 0, & l = 0,1,\ldots,n, \\ \mathbf{1} \cdot \lambda = 1, \\ \lambda, s_l^+, s_l^- \geq 0. \end{cases} \tag{11}$$

The dual problem of (11) is (FDH multiplier form equivalent)

$$\max_{u_l,v_l,z} \quad z$$
$$\text{s.t} \quad \begin{cases} v_l \cdot x_i = 1, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_i\right) - v_l \cdot x_l + z \leq 0, \quad l = 1,2,\ldots,n, \\ u_l \geq \varepsilon, \quad v_l \geq \varepsilon \cdot \mathbf{1}, \quad l = 1,2,\ldots,n. \end{cases} \tag{12}$$

LP problem (12) is the multiplier form equivalent of FDH model. It also reveals the economic meaning of FDH. Coefficients (u, v) are the prices for output y and input x. The profit of DMU i under the price system (u_l, v_l) is calculated by $u_l \cdot y_i - v_l \cdot x_i$, and the input cost of target DMU is normalized $v_l \cdot x_i = 1$. The second constraint of (11) is equivalent to

$$\left(u_l \cdot y_i - z \cdot v_l \cdot x_i\right) - \left(u_l \cdot y_l - v_l \cdot x_l\right) \geq 0, \tag{13}$$

which suggests a nonnegative profit difference between the input-scaled DMU i and DMU l. The upper bound of scale factor z, calculated by letting $i = l$, is $z = 1$, which indicates a scaling down of DMU i's input. FDH scans all the DMUs to find a reference DMU and a price system with the largest scale-down factor z.

Based on the FDH multiplier form equivalent given in (12), we now define the peer-appraisal FDH cross efficiency as follows.

Definition 7 (FDH cross efficiency, benevolent formulation). Given that the FDH efficiency of schedule γ_i is E_{ii}, the FDH cross efficiency E_{ji} of γ_j evaluated by γ_i is the optimal value of z in

$$\max_{u_l,v_l,z} \quad z$$
$$\text{s.t} \quad \begin{cases} v_l \cdot x_j = 1, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_j\right) - v_l \cdot x_l + z \leq 0, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_i\right) - v_l \cdot \left(x_l - E_{ii} \cdot x_i\right) \leq 0, \\ \qquad\qquad l = 1,2,\ldots,n, \\ u_l \geq \varepsilon, \quad v_l \geq \varepsilon \cdot \mathbf{1}, \quad l = 1,2,\ldots,n. \end{cases} \tag{14}$$

Definition 7 is the FDH correspondence of the peer-appraisal CCR (benevolent formulation) in [24], which is reviewed in Section 4.1. The third constraint LP in (14) ensures that the efficiency of DMU i calculated by coefficients (u_l, v_l) is the FDH efficiency E_{ii} (primary goal), and under this constraint, (14) searches for the best efficiency value of E_{ji} (secondary goal).

The following two theorems reveal the relation between FDH efficiency and FDH cross efficiency.

Theorem 8. *The cross efficiency of schedule γ_i evaluated by itself is its FDH efficiency.*

Proof. Assume that the cross efficiency of schedule γ_i evaluated by itself is E_{ii} and the FDH efficiency (simple efficiency) is E.

The calculation of E_{ii} is solving the following LP:

$$\max_{u_l,v_l,z} \quad z$$
$$\text{s.t} \quad \begin{cases} v_l \cdot x_i = 1, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_i\right) - v_l \cdot x_l + z \leq 0, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_i\right) - v_l \cdot \left(x_l - E \cdot x_i\right) \leq 0, \\ \qquad\qquad l = 1,2,\ldots,n, \\ u_l \geq \varepsilon, \quad v_l \geq \varepsilon \cdot \mathbf{1}, \quad l = 1,2,\ldots,n. \end{cases} \tag{15}$$

Compared with the FDH multiplier form equivalent in (12), the LP in (15) has extra constraints of

$$\begin{aligned} & u_l \cdot \left(y_l - y_i\right) - v_l \cdot \left(x_l - E \cdot x_i\right) \\ & \quad = u_l \cdot \left(y_l - y_i\right) - v_l \cdot x_l + E \leq 0, \quad l = 1,2,\ldots,n, \end{aligned} \tag{16}$$

where E is the optimal value of (12). Since $z = E$ satisfies (12), the extra constraints of (16) always hold true. That means E is also the optimal value of (15); thus $E = E_{ii}$. □

Theorem 9. *The cross efficiency E_{ji} of schedule γ_j evaluated by schedule γ_i does not exceed the value of its simple efficiency E_{jj}.*

Proof. It is obvious that the optimal solution in (14) is a feasible solution of calculating schedule γ_j's FDH efficiency using (12)

$$\max_{u_l,v_l,z} \quad z$$
$$\text{s.t} \quad \begin{cases} v_l \cdot x_j = 1, \quad l = 1,2,\ldots,n, \\ u_l \cdot \left(y_l - y_j\right) - v_l \cdot x_l + z \leq 0, \quad l = 1,2,\ldots,n, \\ u_l \geq \varepsilon, \quad v_l \geq \varepsilon \cdot \mathbf{1}, \quad l = 1,2,\ldots,n. \end{cases} \tag{17}$$

Thus the optimal value of z in (14) is not greater than the optimal value of z in (17). □

Using the peer-appraisal FDH proposed in (14), the cross efficiency of a DMU is defined as follows.

Definition 10 (cross efficiency matrix, average cross efficiency, and the most efficient schedule). In a schedule set Γ with n schedules, the cross efficiency matrix is defined by

$$E_{\text{cross}} = \begin{pmatrix} E_{11} & E_{12} & \cdots & E_{1n} \\ E_{21} & E_{22} & \cdots & E_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ E_{n1} & E_{n2} & \cdots & E_{nn} \end{pmatrix}, \tag{18}$$

where E_{ji} is the FDH cross efficiency of γ_j evaluated by γ_i using (14). The average cross efficiency of DMU i is the average value of its cross efficiencies evaluated by all DMUs in Γ, defined by $E_i = \sum_{l=1}^{n} E_{il}/n$. The most efficient schedule (γ_{MES}) is the schedule with the largest average cross efficiency.

The cross efficiency matrix E_{cross} is constructed to calculate the cross efficiency of each DMU. The diagonal elements in E_{cross} are the self-appraisal FDH efficiencies, and the rest of the elements are the peer-appraisal FDH efficiencies. The elements in ith row of E_{cross} are the efficiencies of DMU i rated by peers, and the elements in ith column are the efficiencies of peers rated by DMU i. The cross efficiency of DMU i is the average value of the ith row. And γ_{MES} is the best (both efficient and well metrics balanced) schedule in Γ under our evaluation system.

Corollary 11. *The average cross efficiency E_j of a schedule γ_j, which is defined in Definition 10, is not greater than its simple efficiency E_{jj}.*

Proof. Using the result of previous theorem, the cross efficiency E_{ji} of schedule γ_j evaluated by arbitrary schedule γ_i is not greater than the E_{jj}. Thus E_j, which is the average value of E_{ji}, is not greater than the E_{jj}. □

Then the relation between the FDH efficiency and the most efficient schedule is given in the following theorem.

Theorem 12. *The most efficient schedule of a schedule set Γ is an efficient schedule.*

Proof. Suppose the most efficient schedule is an inefficient schedule, and then there exits an efficient schedule that dominates it. Formally, let $\gamma_\alpha = (x_\alpha, y_\alpha)$ be the most efficient schedule of Γ, where $x_\alpha = (M \quad E \quad L)^T$ is the input vector and $y_\alpha = B$ is the output scalar. Let z_α^* be the cross efficiency of schedule γ_α evaluated by schedule γ_i, and $u_l^* = u_{lb}^*$, $v_l^* = (v_{lM}^* \quad v_{lE}^* \quad v_{lL}^*)$, $l = 1, 2, \ldots, n$, are the optimal coefficients. Let γ_β be a dominant schedule of γ_α.

First assume that γ_β dominates γ_α in output (balance), and $\gamma_\beta = (x_\beta, y_\beta)$, where $x_\beta = x_\alpha = (M \quad E \quad L)^T$ and $y_\beta = B + \Delta B$ has a small increment in balance. Now we calculate the cross efficiency of schedule γ_β evaluated by schedule γ_i as follows:

$$\max_{u_l, v_l, z_\beta} \quad z_\beta$$

$$\text{s.t} \quad \begin{cases} v_l \cdot x_\beta = v_l \cdot x_\alpha = 1, \\ u_l \cdot \left(y_l - y_\beta\right) - v_l \cdot x_l + z_\beta \\ \quad = u_{lB} \cdot \left[y_l - (B + \Delta B)\right] - v_l \cdot x_l + z_\beta \\ \quad = u_l \cdot (y_l - y_\alpha) - v_l \cdot x_l + \left(z_\beta - u_{lB} \cdot \Delta B\right) \le 0, \\ u_l \cdot (y_l - y_i) - v_l \cdot (x_l - E_{ii} \cdot x_i) \le 0, \\ u_l \ge \varepsilon, \quad v_l \ge \varepsilon \cdot \mathbf{1}, \quad l = 1, 2, \ldots, n. \end{cases} \tag{19}$$

Compared with the calculation of the cross efficiency of schedule γ_α evaluated by schedule γ_i, it is easy to verify that $\widehat{z}_\beta = z_\alpha^* + \max_l(u_{lB}^* \cdot \Delta B)$, u_l^*, v_l^*, $l = 1, 2, \ldots, n$, is a feasible solution of the above LP. Then the optimal value of z_β is not less than $z_\alpha^* + u_{lB}^* \cdot \max_l(\Delta B)$, which is greater than z_α^*. That means the cross efficiency of schedule γ_β evaluated by an arbitrary schedule γ_i is greater than the γ_α, which contradicts the assumption of γ_α being the most efficient schedule.

Then assume that γ_β dominates γ_α in input. Without loss of generality, we assume that γ_β dominates γ_α in makespan, and $\gamma_\beta = (x_\beta, y_\beta)$, where $y_\beta = B$ and $x_\beta = (M - \Delta M \quad E \quad L)^T$ has a small decrement in makespan. Then the cross efficiency of schedule γ_β evaluated by schedule γ_i is calculated by solving the following LP:

$$\max_{u_l, v_l, z_\beta} \quad z_\beta$$

$$\text{s.t} \quad \begin{cases} v_l \cdot x_\beta = v_{lM} \cdot (M - \Delta M) + v_{lE} \cdot E + v_{lL} \cdot L = 1, \\ u_l \cdot \left(y_l - y_\beta\right) - v_l \cdot x_l + z_\beta \\ \quad = u_{lB} \cdot \left(y_l - y_\beta\right) \\ \qquad - \left(v_{lM} \cdot x_{lM} + v_{lE} \cdot x_{lE} + v_{lL} \cdot x_{lL}\right) + z_\beta \le 0, \\ u_l \cdot \left(y_l - y_i\right) - v_l \cdot \left(x_l - E_{ii} \cdot x_i\right) \\ \quad = u_l \cdot \left(y_l - y_i\right) \\ \qquad - \left[v_{lM} \cdot \left(x_{lM} - E_{ii} \cdot x_{iM}\right)\right. \\ \qquad\quad + v_{lE} \cdot \left(x_{lE} - E_{ii} \cdot x_{iE}\right) \\ \qquad\quad \left. + v_{lL} \cdot \left(x_{lL} - E_{ii} \cdot x_{iL}\right)\right] \le 0, \\ u_l \ge \varepsilon, \quad v_l \ge \varepsilon \cdot \mathbf{1}, \quad l = 1, 2, \ldots, n. \end{cases} \tag{20}$$

Let $v_l' = (v_{lM}' \quad v_{lE}' \quad v_{lL}') = (v_{lM}^*/(1 + \Delta M \cdot v_{lM}^*))(v_{lM}^* \quad v_{lE}^* \quad v_{lL}^*)$, $u_l' = (v_{lM}^*/(1 + \Delta M \cdot v_{lM}^*))u_l^*$, and $z_\beta' = z_\alpha^* \cdot (1 + \Delta M \cdot \max_l(v_{lM}^*))$. By substituting v_l', u_l', and z_β' in (20), it is easy to verify that they are a feasible solution of (20). Then the optimal solution of the optimal value of z_β is not less than $z_\alpha^* \cdot (1 + \Delta M \cdot \max_l(v_{lM}^*))$, which contradicts the assumption of γ_α being the most efficient schedule.

Finally, if γ_β dominates γ_α in multiple metrics, say $\gamma_\beta = (M - \Delta M \quad E \quad L \quad B + \Delta B)$, intermediary schedules of $\gamma_i = (M - \Delta M \quad E \quad L \quad B)$ and $\gamma_i = (M - \Delta M \quad E \quad L \quad B + \Delta B)$ can be constructed to prove that γ_α is not the most efficient schedule using the previous results. □

Now we will prove the unit invariance property of our peer-appraisal FDH proposal as well as the original FDH model.

Theorem 13 (unit invariance property). *The values of optimal goal in* (5) *and* (12) *are independent of the units that inputs and outputs are measured in.*

Proof. First we prove that FDH efficiency (5) is unit invariant. The FDH model in (5) is equivalent to the LP in (11). So if (11) is unit invariant, then (5) is too.

In a schedule set $\Gamma = \{\gamma_l = (x_l, y_l) \mid l = 1, 2, \ldots, n\}$, let $x_l = (M_l \quad E_l \quad L_l)^T$ and $y_l = B_l$ be the input vector and output scalar, and let z^*, $u_l^* = (u_{lM}^* \quad u_{lE}^* \quad u_{lL}^*)$ and v_l^* be the optimal value for (12).

Under the new unit system, the inputs and outputs now are the original values multiplied by conversion coefficients, denoted by $x_l' = (M_l' \quad E_l' \quad L_l')^T = (\alpha_M M_l \quad \alpha_E E_l \quad \alpha_L L_l)^T$ and $y_l' = \beta y_l$. Then applying (11) to the new schedule set $\Gamma' = \{\gamma_l' = (x_l', y_l') \mid l = 1, 2, \ldots, n\}$, obviously we have a feasible solution of $\widehat{z}' = z^*$, $\widehat{u}_l' = (u_{lM}^*/\alpha_M \quad u_{lE}^*/\alpha_E \quad u_{lL}^*/\alpha_L)$ and $\widehat{v}_l' = v_l^*/\beta$, which is transformed from the original problem. Therefore, we have the optimal value of new problem $z'^* \geq \widehat{z}'$. Suppose $z'^* > \widehat{z}'$, and the weight coefficients are $u_l'^* = (u_{lM}'^* \quad u_{lE}'^* \quad u_{lL}'^*)$ and $v_l'^*$ under this circumstance. Converting the weight coefficients to the original unit system, we have $\widehat{u}_l = (\alpha_M u_{lM}'^* \quad \alpha_E u_{lE}'^* \quad \alpha_L u_{lL}'^*)$ and $\widehat{v}_l = \beta v_l'$, which also satisfy the constraints of original problem in (12). This suggests that z'^*, $\widehat{u}_l$, and $\widehat{v}_l$ are also a feasible solution of original problem, and $z'^* > \widehat{z}' = z^*$ contradicts the optimal assumption of z^*. So $z'^* = \widehat{z}'$ is the only possibility. This proves that the FDH efficiency is not affected by the units that inputs and outputs are measured in.

Using this result, the unit invariance of peer-appraisal FDH can be proved in the same manner. □

5. Efficient Scheduling Using CrosFDH-GA

5.1. Basic Design. The most straightforward way to introduce our FDH cross evaluation method to the genetic algorithm (GA) is to calculate the average cross efficiency of all individuals in the pool and use the efficiency value as the fitness of each individual. The problem of this simple solution is the high computational requirement of DEA calculation.

For a genetic algorithm with 1000 population, the calculation of FDH simple efficiency of a single individual is to solve a LP with 4001 variables (u_{lB}, v_{lM}, v_{lE}, v_{lL}, $l = 1, 2, \ldots, 1000$, and z) and 2000 constraints using (12), and the calculation of FDH simple efficiency of all individuals is to solve 1000 such LPs.

Then the calculation of FDH cross efficiency of an individual evaluated by another individual is to solve a LP with 4001 variables and 3000 constraints using (14). To calculate the average cross efficiency of an individual, solving 999 such LPs is required. And to calculate the average cross efficiency of all individuals, it requires repeating the process for 1000 times.

In our computing configuration (Intel i5-3210M, 4 Gb RAM, 32 bit Win7, VS2010, and GLPK 4.47), the calculation of a single FDH simple efficiency and a single FDH cross efficiency in the above DEA-GA implementation takes about 3.6 seconds and 6.2 seconds on average. The calculation of an average cross efficiency requires over 10 minutes. The DEA solving time of 1000 individuals in a single generation is estimated to be about 7 days. If the GA runs for 50 generations, the whole solving time is near a year, which is beyond acceptable level.

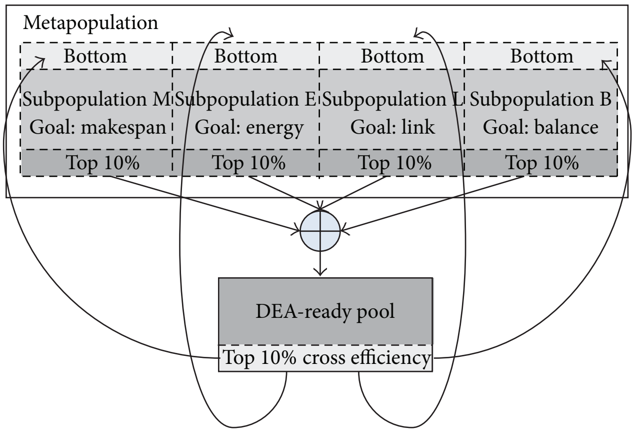


Figure 2: The Subpopulation updating process.

In this paper, we propose a solution to this problem using a "divide-and-conquer" method. The whole population (*Metapopulation*) is divided in to 4 subpopulations: *Subpopulation M*, *Subpopulation E*, *Subpopulation L*, and *Subpopulation B*, each of which experiences its own evolution towards a single optimization goal (*Makespan*, *Energy*, *average Link load*, and *workload Balance*, correspondingly). In each generation, after the algorithm evaluates the performance of every individual, the elites of each subpopulation are selected and regrouped as the *DEA-ready pool* for the DEA evaluation process. The basic idea is that, according to Theorem 4, the more preeminent a schedule is in one metric, the less likely it is dominated by another schedule. Then the top performers in the DEA-ready pool are duplicated to each subpopulation and replace the bottom individuals, and the subpopulations continue to evolve. The process is shown in Figure 2.

5.2. Genetic Operations. In our proposal, a chromosome or an individual represents one schedule solution. The structure of a chromosome is an array with the size of processors number, and the value of its element, $chr[i] = j$, represents that task i is assigned to processor j.

The evolution of the four subpopulations is independent, each of which goes through a complete series of genetic operations. The basic framework of genetic algorithm is based on the proposal in [27] as follows.

Step 1 (initialization). The chromosome is randomly created and added to the subpopulation. When the number of population reaches *subpopulation_size*, algorithm goes to the next step.

Step 2 (evaluation). Performance of each individual in the pool is evaluated.

Table 1: Two randomly generated chromosomes.

Chromosome	Index								
	1	2	3	4	5	6	7	8	9
chr1	3	1	3	2	2	3	1	2	3
chr2	2	2	1	3	1	1	1	1	3

Step 3 (selection). Chromosomes are ordered according to their subpopulation's optimization goal, and the top *sel_ratio* chromosomes directly enter the next generation's pool.

Step 4 (crossover). Two random chromosomes, *chr1* and *chr2*, are selected from current pool, and two new chromosomes, *nchr1* and *nchr2*, are generated by swapping middle part of the chromosome array. This step produces population of *cros_ratio* $*$ *subpopulation_size*.

Step 5 (mutation). A random chromosome is picked, and values of two random positions of the chromosome are swapped to produce a new chromosome. The population generated in the mutation step accounts for *mut_ratio* of *subpopulation_size*.

Step 6 (termination). GA is terminated after certain number of generations. If GA does not meet its terminal condition, the algorithm iterates to Step 2 and repeats the whole process.

For example, consider a scheduling problem of 9 tasks (task 1~task 9) scheduling to 3 processors (processor 1~ processor 3). Two randomly generated chromosomes, *chr1* and *chr2*, are listed in Table 1. Following the previous definition, *chr1* represents that task 2 and task 7 are scheduled to processor 1; tasks 4, 5, and 8 are scheduled to processor 2; and tasks 1, 3, 6, and 9 are scheduled to processor 3. Chromosome *chr2* represents that tasks 3, 5, 6, 7, and 8 are scheduled to processor 1; task 1 and task 2 are scheduled to processor 2; and task 4 and task 9 are scheduled to processor 3.

In the *crossover* operation, two randomly selected chromosomes, *chr1* and *chr2*, swap their middle part of chromosome to form two new chromosomes *nchr1* and *nchr2*, as shown in Figure 3.

In the mutation operation, a new chromosome *nchr1* is generated by randomly picking a chromosome *chr1*, and swapping two arbitrary positions in *chr1* as shown in Figure 4.

5.3. Cross Evaluation of Individuals.

5.3. Cross Evaluation of Individuals. The elites of each subpopulation are selected to form a *DEA-ready pool*. Then the DEA approach is applied to the individuals in this pool.

First, the FDH simple efficiency of every individual in the pool is calculated, and the inefficient individuals are removed. According to Corollary 5, the removal of inefficient DMUs does not change the dominant position of the efficient ones. Then the average cross efficiencies of the remaining individuals are calculated, and the individual with the largest value of average cross efficiency is marked as the most efficient schedule.

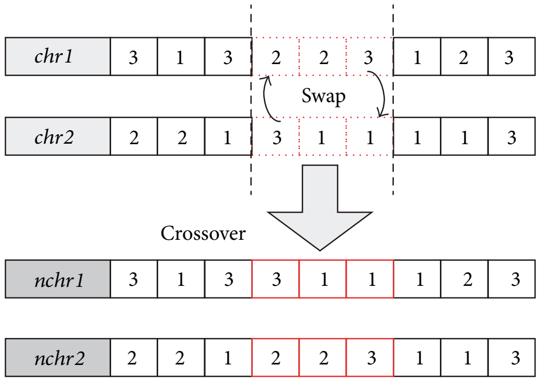


Figure 3: The crossover operation.

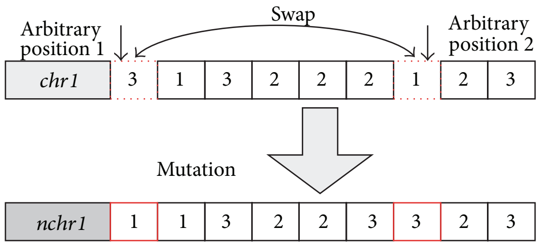


Figure 4: The mutation operation.

The reason for the removal of inefficient DMUs is threefold: first of all, as proven in Theorem 12, we know that the most efficient schedule which we are pursuing is not an inefficient schedule; secondly, the removal of inefficient schedules eliminates the influence of these obviously defected schedules on the coefficients of the following calculation of FDH cross efficiency (otherwise, an inefficiency schedule would become a constraint in the calculation of FDH cross efficiency according to (14)); finally, it further reduces the computational demand of our algorithm.

Pseudocode 1 shows cross evaluation process in our algorithm.

6. Computational Experiments and Discussion

6.1. Simulation Results of FDH Cross Evaluation Formulation. In this section, we extract 40 actual schedule solutions from our simulation, and use our proposed FDH cross evaluation, as well as other DEA methods, to analyze the performance of these DMUs. The chosen schedules are from the 5th generation of our CrosFDH-GA for a schedule problem of scheduling 100 tasks onto a 4 × 4 mesh NoC. Top 10 best schedules in each subpopulation are grouped as our schedule set. The schedules are listed in Table 2, and for more intuitive observation, all the metrics shown are preprocessed by dividing the value by the average value of each metric in the set. As we have proved in Theorem 13, this normalization process will not affect the value of the DMU's efficiency.

CCR efficiency (CCR), BCC efficiency (BCC), FDH efficiency (FDH), CCR super efficiency (Super CCR), CCR average Cross efficiency (Cros CCR), and FDH average Cross

```
BEGIN SCHEDULE-EVALUATION
  (1) FOR each schedule $\gamma_l$ in the DEA-ready pool $\Gamma$ DO:
      (1.1) Calculate $E_{ll}$ using FDH DEA;
  (2) END FOR
  (3) Regroup efficient schedule ($E_{ll} = 1$) as $\Gamma_e$;
  (4) FOR each schedule $\gamma_j$ in $\Gamma_e$;
      (4.1) FOR each schedule $\gamma_i \neq \gamma_j$ in $\Gamma_e$ DO:
          (4.1.1) Calculate the peer-appraisal FDH $E_{ji}$;
      (4.2) END FOR;
      (4.3) Calculate the average cross efficiency $E_j$ of $\gamma_j$;
  (5) END FOR
  (6) Sorting the cross efficiency of schedules in $\Gamma_e$;
  (7) Mark the schedule with the largest average cross efficiency as $\gamma_{\mathrm{MES}}$;
END SCHEDULE-EVALUATION
```

Pseudocode 1: Pseudocode of cross evaluation of DEA-ready pool.

efficiency (Cros FDH) of each DMU are also calculated and listed in Table 2. All the DEA formulations are solved using GLPK [28].

Two kinds of FDH cross efficiencies, Cros FDH (All) and Cros FDH (Eff), are presented in Table 2. The difference between these two is as follows. Cros FDH (All) calculates the FDH cross efficiency using all 40 schedules while Cros FDH (Eff) only takes into account the FDH efficient ones, which is suggested in Section 5.3. The value of CrosFDH (Eff) for inefficient schedules is not calculated and marked with "—."

Moreover, a *maverick index* (MI), which is suggested in [24], is calculated for each FDH cross efficiency. MI for DMU i is calculated by $\mathrm{MI}_i = (E_{ii} - E_i)/E_{ii}$.

MI measures the difference between a DMU's simple efficiency and its cross efficiency. The larger MI value implies that the DMU is more likely to be a maverick that "cheats" a high simple efficiency by choosing a particular set of coefficients that favors its only strength and depreciating other metrics, in our words, a metric-unbalanced DMU.

From Table 2 we observe that, among the 3 simple efficiency formulations, CCR has the smallest efficient schedule number of 2, while BCC has 9 efficient schedules, and FDH has 13. Both efficient schedules in CCR are efficient in BCC and FDH; all 9 efficient schedules in BCC are also FDH efficient, and FDH has 4 extra schedules: *schedule 4*, *schedule 15*, *schedule 18*, and *schedule 19*, than BCC. The difference between the efficient schedule numbers of these three DEA models is caused by the different shapes of efficient frontier, which is generated by different constraints in the model formulations. The convex-shaped efficient frontier of BCC in the multi-input multioutput space contains more DMUs than the CCR efficient frontier, while the staircase-like FDH efficient frontier has the largest number of efficient DMUs on it. Moreover, the value of CCR efficiency of a schedule is generally the smallest one among the three models, and BCC is generally smaller than the FDH efficiency. In fact, as pointed out in [29], the FDH efficiencies are generally higher than CCR and BCC efficiencies.

Three DMU ranking methods, CCR super efficiency, CCR cross efficiency, and FDH cross efficiency, are also compared in Table 2. As observed in the table, CCR super efficiency and CCR cross efficiency are consistent with the CCR simple efficiency, and both CCR ranking methods mark *schedule 32* as the best schedule. For the FDH average cross efficiency, it is easy to tell that CrosFDH (all) is more discriminating than the previous methods. The FDH average cross efficiencies of 13 FDH efficient schedules vary from 0.9934 (*schedule 20*, ranks 1 in 40 schedules) to 0.905 (*schedule 4*, ranks 35 in 40 schedules). The reason for an efficient *schedule 4* achieving such a low average cross efficiency score is explained by its MI value. *schedule 4* has a high MI of 0.105, which suggests that it "cheats" in the FDH simple efficiency calculation. A close look on its metrics reveals that it compromises too much on the E and L metric. The same situation happens with *schedule 31* (ranks 25 in 40) and *schedule 32* (ranks 24 in 40), in which both are classified as CCR efficient schedules and *schedule 32* even being marked as the "best" schedule under CCR super efficiency and CCR cross efficiency analysis. MIs of *schedule 31* and *32* are 0.0568 and 0.0553; the relevantly high MIs imply that they are more likely to be mavericks. And after examining their metrics, it is shown that they are both metric unbalanced because they trade too much performance on the E and L metric for the B metric.

As to the two different implementations of the FDH average cross efficiency, the Cros FDH (All) and Cros FDH (Eff), deliver very similar results on the DMU ranking. Top 3 schedules under Cros FDH (All) are *schedules 20*, *17*, and *18*. Comparing to the top 3 ranking of Cros FDH (Eff), *schedules 17*, *20*, and *18*, only a slight change of order exists, which validates the process of inefficient DMU removal in the cross evaluation of individuals proposed in Section 5.3.

Moreover, the average values of four metrics and MI (All) in each subpopulation are calculated and listed in Table 3. The most interesting results are in the *Subpopulation B*. The average *workload balance* of *Subpopulation B* is almost as twice as the average value of the rest three subpopulations. However, the performance of other metrics is not so good in the *Subpopulation B* and is 1.2%, 15.4%, and 16% larger than the average value of the rest three subpopulations in *makespan*, *energy*, and *average link load*, respectively. This

Table 2: DEA analysis results of 40 sample schedules.

Schedule	Subpop.	Makespan (input 1)	Energy (input 2)	Link (input 3)	Balance (output)	CCR	BCC	FDH	SuperCCR	CrosCCR	CrosFDH (All)	MI (All)	CrosFDH (Eff)	MI (Eff)
1	*M*	0.9478	0.9695	0.9368	0.9657	0.6624	1.0000	1.0000	0.6624	0.6503	0.9878	0.0123	0.9917	0.0084
2	*M*	0.9650	1.0624	1.0396	0.8970	0.5545	0.9821	0.9821	0.5545	0.5497	0.9371	0.0480	—	—
3	*M*	0.9683	0.9993	1.0154	0.7146	0.4522	0.9788	0.9788	0.4522	0.4492	0.9484	0.0320	—	—
4	*M*	0.9693	1.1701	1.1363	1.4241	0.8520	0.9933	1.0000	0.8520	0.8065	0.9050	0.1050	0.9123	0.0961
5	*M*	0.9696	1.0624	1.0295	0.9227	0.5759	0.9775	0.9775	0.5759	0.5694	0.9368	0.0435	—	—
6	*M*	0.9732	1.0587	1.0557	0.6595	0.4014	0.9739	0.9739	0.4014	0.3991	0.9245	0.0534	—	—
7	*M*	0.9732	1.0736	1.0738	0.6453	0.3862	0.9739	0.9739	0.3862	0.3847	0.9184	0.0605	—	—
8	*M*	0.9748	1.1070	1.0759	0.9227	0.5511	0.9723	0.9723	0.5511	0.5472	0.9181	0.0590	—	—
9	*M*	0.9758	0.9955	0.9973	0.7551	0.4865	0.9721	0.9739	0.4865	0.4813	0.9488	0.0265	—	—
10	*M*	0.9758	0.9250	0.9429	0.7274	0.4957	0.9953	0.9963	0.4957	0.4889	0.9822	0.0144	—	—
11	*E*	1.0048	0.9064	0.9368	0.6002	0.4143	0.9918	0.9918	0.4143	0.4055	0.9649	0.0279	—	—
12	*E*	1.0048	0.9175	0.9288	0.6002	0.4143	0.9828	0.9918	0.4143	0.4055	0.9649	0.0279	—	—
13	*E*	1.0048	0.9138	0.9469	0.5826	0.3989	0.9839	0.9841	0.3989	0.3900	0.9611	0.0239	—	—
14	*E*	1.0103	0.9138	0.9409	0.5487	0.3757	0.9837	0.9837	0.3757	0.3689	0.9573	0.0276	—	—
15	*E*	1.0482	0.9138	0.9348	0.7274	0.5000	0.9875	1.0000	0.5000	0.4900	0.9490	0.0538	0.9704	0.0305
16	*E*	1.0048	0.9138	0.9550	0.5759	0.3943	0.9839	0.9841	0.3943	0.3830	0.9599	0.0252	—	—
17	*E*	0.9745	0.9138	0.9187	0.9227	0.6454	1.0000	1.0000	0.6454	0.6340	0.9929	0.0072	0.9995	0.0005
18	*E*	0.9745	0.9212	0.9227	0.9507	0.6620	0.9979	1.0000	0.6620	0.6504	0.9915	0.0086	0.9985	0.0015
19	*E*	0.9761	0.9212	0.9248	0.9507	0.6606	0.9966	1.0000	0.6606	0.6504	0.9915	0.0086	0.9985	0.0015
20	*E*	0.9722	0.9212	0.9227	1.0153	0.7070	1.0000	1.0000	0.7070	0.6947	0.9934	0.0066	0.9992	0.0008
21	*L*	1.0488	0.9287	0.9066	0.8850	0.6273	1.0000	1.0000	0.6273	0.6096	0.9631	0.0383	0.9896	0.0105
22	*L*	0.9888	0.8990	0.9127	0.6695	0.4714	1.0000	1.0000	0.4714	0.4628	0.9819	0.0184	0.9903	0.0098
23	*L*	1.0087	0.9584	0.9147	0.9657	0.6784	1.0000	1.0000	0.6784	0.6601	0.9755	0.0251	0.9927	0.0074
24	*L*	0.9729	0.9287	0.9147	0.5965	0.4191	1.0000	1.0000	0.4191	0.4106	0.9821	0.0183	0.9937	0.0063
25	*L*	1.0488	0.9435	0.9227	0.8734	0.6082	0.9850	0.9956	0.6082	0.5921	0.9550	0.0426	—	—
26	*L*	0.9791	0.9435	0.9227	0.8734	0.6082	0.9950	0.9956	0.6082	0.5955	0.9835	0.0124	—	—
27	*L*	1.0133	0.9175	0.9227	0.8412	0.5858	0.9924	0.9960	0.5858	0.5739	0.9701	0.0266	—	—
28	*L*	1.0133	0.9250	0.9248	0.9507	0.6606	0.9932	0.9978	0.6606	0.6468	0.9743	0.0242	—	—
29	*L*	0.9729	0.9435	0.9288	0.6320	0.4373	0.9929	0.9993	0.4373	0.4289	0.9771	0.0227	—	—
30	*L*	1.0495	0.9510	0.9288	0.7209	0.4988	0.9785	0.9892	0.4988	0.4857	0.9414	0.0507	—	—
31	*B*	0.9709	1.0550	1.0759	1.6743	1.0000	1.0000	1.0000	1.0013	0.9991	0.9463	0.0568	0.9535	0.0488
32	*B*	0.9722	1.0475	1.0759	1.6743	1.0000	1.0000	1.0000	1.0071	0.9999	0.9476	0.0553	0.9550	0.0471
33	*B*	1.0504	1.1664	1.1665	1.5996	0.8831	0.9220	0.9243	0.8831	0.8785	0.8682	0.0646	—	—
34	*B*	1.0103	1.1590	1.1665	1.5996	0.9181	0.9586	0.9610	0.9181	0.8843	0.8848	0.0861	—	—
35	*B*	1.0221	1.1367	1.1403	1.5996	0.9076	0.9476	0.9499	0.9076	0.8996	0.8907	0.0666	—	—
36	*B*	1.0279	1.0698	1.0859	1.5996	0.9465	0.9747	0.9907	0.9465	0.9408	0.9196	0.0774	—	—
37	*B*	1.0279	1.0698	1.0879	1.5341	0.9061	0.9590	0.9889	0.9061	0.9009	0.9185	0.0766	—	—
38	*B*	1.0504	1.1590	1.1585	1.5341	0.8510	0.9199	0.9287	0.8510	0.8477	0.8713	0.0659	—	—
39	*B*	1.0504	1.1293	1.1202	1.5341	0.8800	0.9314	0.9604	0.8800	0.8730	0.8893	0.0800	—	—
40	*B*	1.0534	1.0996	1.0799	1.5341	0.9129	0.9661	0.9963	0.9129	0.9011	0.9078	0.0975	—	—

Table 3: The average value of metrics of each subpopulation.

Subpopulation	Average makespan	Average energy	Average link load	Average balance	Average MI (All)
M	0.9618	0.9473	0.9399	0.8465	0.0455
E	0.9885	0.9138	0.9298	0.8077	0.0217
L	1.0491	0.9398	0.9177	0.8029	0.0279
B	1.0121	1.0773	1.0779	1.6042	0.0727

phenomenon indicates that the optimization on the B metrics is in conflict with other metrics, especially with E and L metrics. The assertion is also supported by the MI (All). The average MI in *Subpopulation B* is 129.3% larger than the average value of the rest three subpopulations, which suggests that the schedules in *Subpopulation B* are extremely metric unbalanced, compared with the schedules in other subpopulations. Thus, in order to achieve a highly metric-balanced schedule solution, compromise must be made on the B metric.

6.2. Simulation Results of CrosFDH-GA Scheduling Algorithm

6.2.1. Simulation Setup. In this section, comparative simulations are made to evaluate the performance of our CrosFDH-GA scheduling algorithm. Twenty DAGs, *tg1*~*tg20*, are generated using *Task Graph For Free* (TGFF) [30]. The task number of generated DAGs varies from 50 to 100. Along with two real-world application, solving *laplace equation* using Gauss-Seidel algorithm [31] and *molecular dynamic coding* [32], a total of 22 task sets are simulated.

The control groups of our simulation are four GAs with different global objective functions, and they are multiplication and division (MD), weighted sum (WS), weighted exponential sum (WES), and exponential weighted criterion (EWC) based on global criterion method [33]. The fitness functions are presented as follows:

$$\begin{aligned}
\text{fitness}_{\text{MD}} &= M^{-1} \cdot E^{-1} \cdot L^{-1} \cdot B, \\
\text{fitness}_{\text{WS}} &= w_M M^{-1} + w_E E^{-1} + w_L L^{-1} + w_B B, \\
\text{fitness}_{\text{WES}} &= w_M M^{-2} + w_E E^{-2} + w_L L^{-2} + w_B B^2, \qquad (21)\\
\text{fitness}_{\text{EWC}} &= \left(e^{w_M} - 1\right) \cdot e^{M^{-1}} + \left(e^{w_E} - 1\right) \cdot e^{E^{-1}} \\
&\quad + \left(e^{w_L} - 1\right) \cdot e^{L^{-1}} + \left(e^{w_B} - 1\right) \cdot e^{B},
\end{aligned}$$

where w_M, w_E, w_L, and w_B are the weight coefficients of the corresponding metric. In our implementation, w_M, w_E, w_L, and w_B are all set to 0.25, which means there is no preference between four metrics.

Moreover, the *sel_ratio*, *cros_ratio*, and *mut_ratio* of GA are 0.2, 0.4, and 0.4, and GA terminates its iteration after 50 generations. The population of four global objective function-based GAs is 1000. The *subpopulation_size* of CrosFDH-GA is 250, which ensures that total population is 1000 individuals.

All the output schedules are simulated under a System C based cycle-accurate NoC simulator, which is a wormhole-routing modified version of [34]. The implemented NoC simulator is a 4 × 4 mesh NoC with the router structure illustrated in Figure 3. The link width is 16 bit, and the FIFO depth is one flit. XY routing is used to forward packets and RR arbitration is adopted to solve contentions. The NoC simulator also integrates Orion 2.0 [35] to measure the actual routing energy during execution of a DAG.

6.2.2. Results and Discussion. The simulation results of 22 task sets, which consist of 20 DAGs generated by TGFF and two real-world applications of solving Laplace equation (LE) using Gauss-Seidel algorithm and molecular dynamic coding (MDC), under four global criterion GAs and our CrosFDH GA, are illustrated in Figure 5. All the shown makespan and energy metrics are the actual results measured in our NoC simulator.

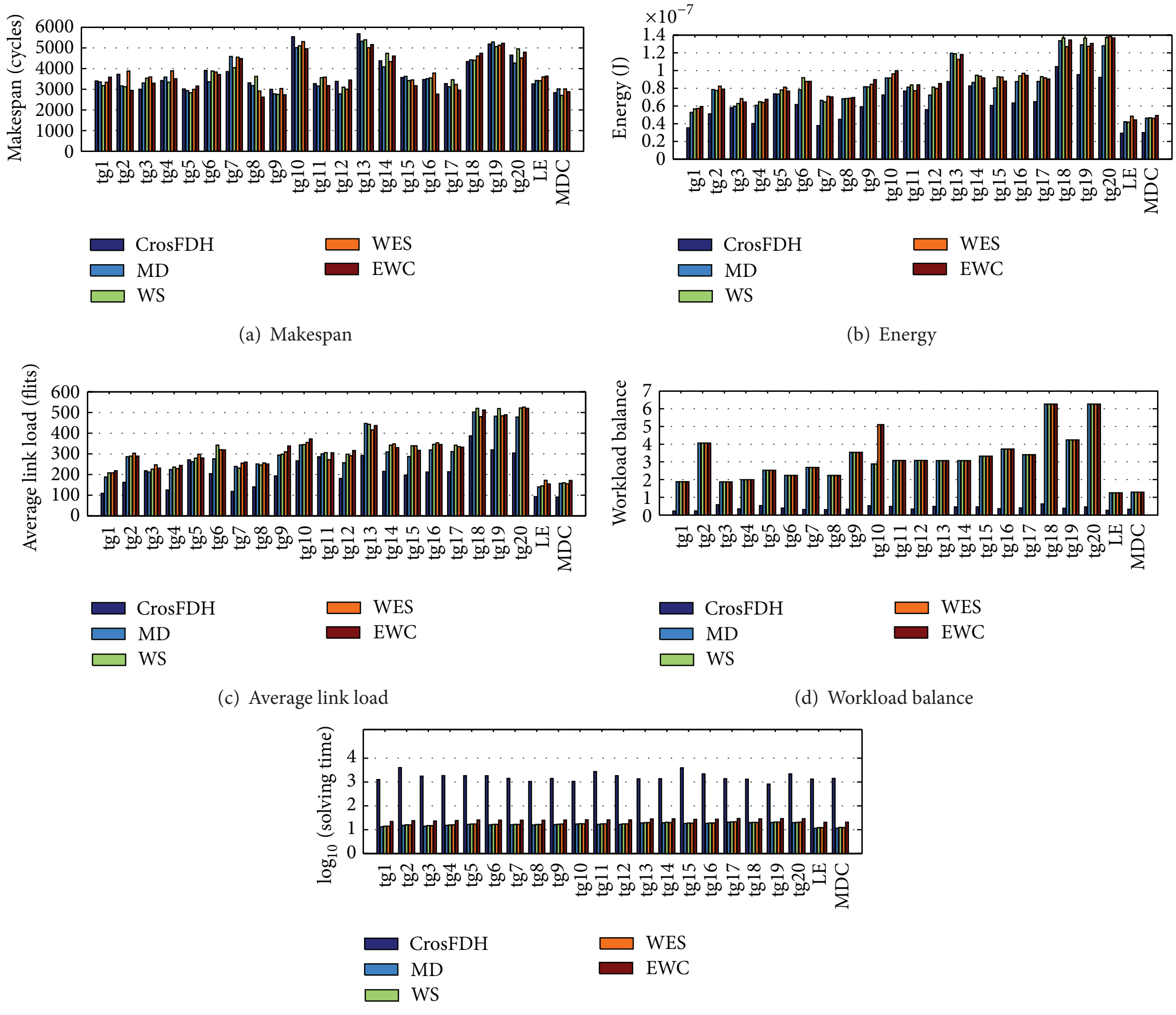
As shown in Figure 5, it is observed that the four global criterion GAs render similar performance on the M, E, L, and B metrics, while the proposed CrosFDH GA exhibits its different tendency on the optimization of metrics. As shown in Figure 5(a), all five scheduling algorithms demonstrate same level of performance on the *makespan*. The main performance difference exists in the optimization of the energy (Figure 5(b)), average link load (Figure 5(c)), and workload balance metrics (Figure 5(d)). The four global criterion GAs always output the schedule solution with the best *workload balance* (in *tg10*, MD-GA and WS-GA do not find the schedule solution with the best B metric within 50 generations as the WES-GA and EWC-GA do), as shown in Figure 5(d). On the other hand, our proposal always compromises on the B metrics and trades for better optimization on the E and L metrics.

To be more specific, we normalize the M, E, L, and B metrics of the five scheduling algorithms to the CrosFDH correspondence in 22 task sets and calculate the average value of these metrics for each scheduling algorithm. The results are shown in Table 4.

From the table, the four global criterion GAs have eight times as much B metric as CrosFDH-GA on average. However, our proposal has better performance on both E and L metrics. The average energy of our algorithm is about 36% smaller, and the average link load is about 29% smaller than the rest algorithms on average. This tendency of optimization in the CrosFDH-GA is explained by the previous analysis of schedule's FDH cross efficiency, which suggests that a better metric-balanced schedule should trade the B metric for the E and L metrics.

Moreover, in our observation, schedule solutions that show similar performance to the output schedule of the four global criterion GAs exists in the *Subpopulation B* of the CrosFDH-GA. However, these schedule solutions are depreciated during the peer-appraising process of FDH cross efficiency, which supports the conclusion that the output schedules of the global criterion GAs are metric unbalanced.

Figure 5(e) illustrates the solving time (measured in seconds) of five algorithms, and for illustrative presentation,



(a) Makespan

(b) Energy

(c) Average link load

(d) Workload balance

(e) Solving time

Figure 5: The comparative results of CrosFDH, MD, WS, WES, and EWC-GA.

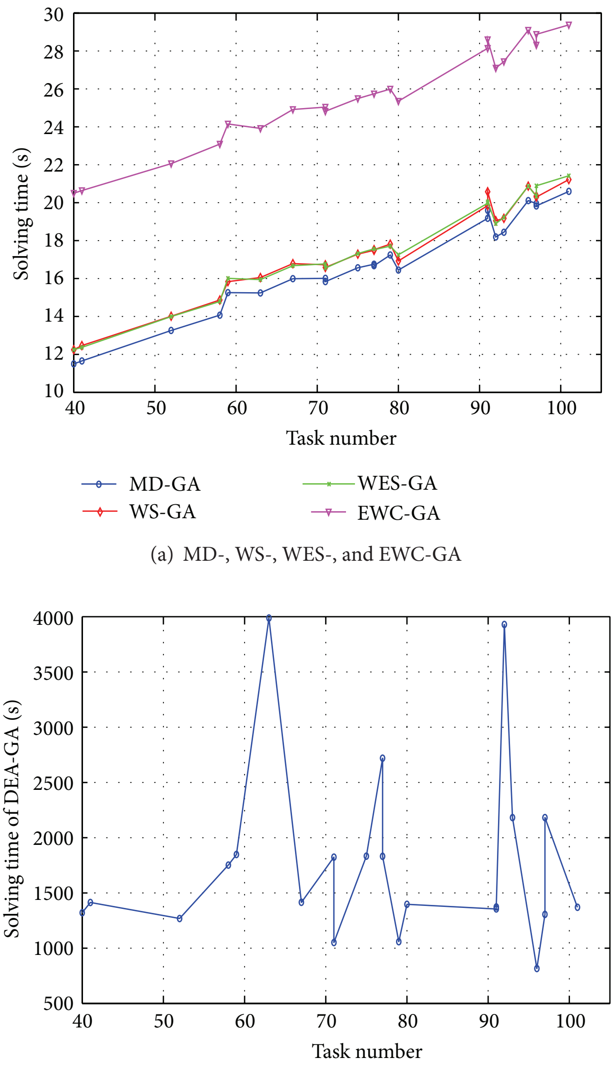
Table 4: The average value of metrics of each scheduling algorithm.

Algorithms	Average makespan	Average energy	Average link load	Average balance
CrosFDH-GA	1.0000	1.0000	1.0000	1.0000
MD-GA	0.9776	1.3451	1.4852	8.0536
WS-GA	0.9992	1.4074	1.5757	8.0536
WES-GA	1.0213	1.4214	1.5883	8.2483
EWC-GA	0.9786	1.4323	1.6036	8.2483

the data in Figure 5(e) are preprocessed by applying $\log_{10}$ to the solving time of each algorithm. According to the figure, MD-GA has the smallest solving time in the five algorithms. The WS-GA, WES-GA, and EWC-GA have 4.4%, 4.5%, and 54.6% more solving times than the MD-GA on average. The CROSFDH-GA has the largest solving time which is nearly 109 times larger than MD-GA. The reason of such long solving time is caused by the high computation load introduced by DEA analysis. In the CROSFDH-GA, 90.2% of the solving time is used to calculate the average FDH cross efficiency, 8.3% of the solving time is used to calculate FDH simple efficiency, and the rest of the algorithm consumed only 1.5% of the solving time.

In MD-GA, WS-GA, WES-GA, and EWC-GA, a clear increasing trend of solving time is observed as the scale of the scheduling problem (task number) rises, as shown in Figure 6(a). However such trend is not observed in CROSFDH-GA, as shown in Figure 6(b).

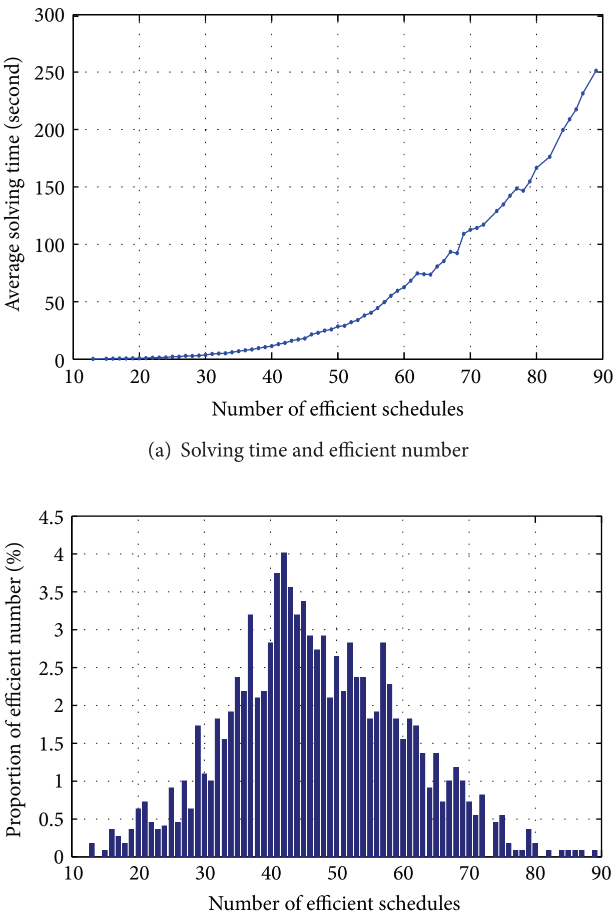
The reason of this phenomenon in CROSFDH-GA is that the solving time of CROSFDH-GA is largely determined by the calculation time of average FDH cross efficiency, and the calculation time of average FDH cross efficiency is depending on the size of the schedule set, which in our CROSFDH-GA is the number of efficiency schedules in the DEA-ready pool in each generation. Thus, the solving time of CROSFDH-GA is not directly related to the task number of a schedule problem



(a) MD-, WS-, WES-, and EWC-GA

(b) CROSFDH-GA

Figure 6: The relation of solving time and task number.



(a) Solving time and efficient number

(b) Efficient number distribution

Figure 7: Efficient schedule number in CROSFDH-GA.

but to the number of efficient schedules in the DEA-ready pool in each generation.

Figure 7(a) shows the relation between number of efficient schedules in the DEA-ready pool and the average solving time of the average FDH cross efficiency in a generation. As shown in the figure, the solving time rises rapidly as the efficient schedule number increases. Moreover, Figure 7(b) gives a statistic result of efficient schedule number during all 22 simulations. As observed in the figure, most of the generations during the simulation have the efficient schedule number that lies between 30 and 70, which requires about 7 to 100 seconds average FDH cross efficiency solving time.

Finally, Figure 8 demonstrates the trend of how four metrics converge during the iteration. The illustrated metrics are the best ones in the corresponding subpopulations in each generation and are all normalized to the final value of the 50th generation. Figure 8 shows that the M metric is the first converged metric, followed by the E metric and L metric, which both converge in the same pace. And the B metric is the last converged metric.

7. Conclusion

In this paper, a FDH cross efficiency formulation, as well as a CrosFDH-GA algorithm, is proposed for the task scheduling problem on the NoC-based MPSoC. Four common metrics, namely, makespan, routing energy, average link load, and workload balance, are used to construct the multi-input multioutput DMU model. After using FDH simple efficiency to eliminate inefficient (dominated) schedules, the peer-appraisal FDH cross efficiency is introduced to ranking schedules, during which the maverick (metric-unbalanced) schedules are depreciated. Then a FDH cross efficiency-based genetic algorithm with four subpopulations, each of which optimizes a single metric, is proposed for solving actual scheduling problem on NoC. According to our simulation results, the proposed FDH cross efficiency effectively distinguishes the schedule solutions according to the balance of their metrics, and our CrosFDH always outputs more metric-balanced schedules than other global criterion GAs.

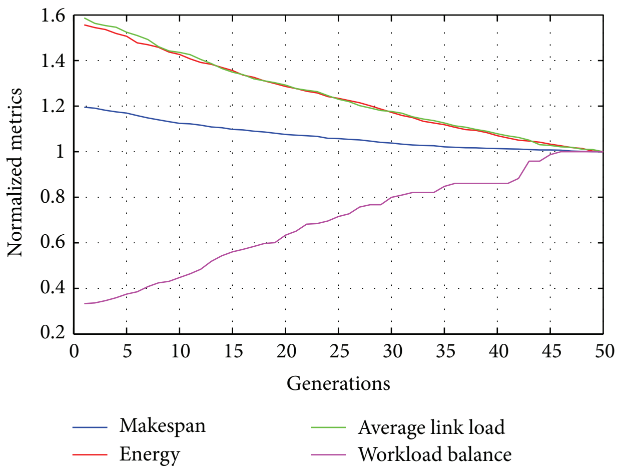


Figure 8: The trend of convergence of four metrics.

Conflict of Interests

The authors certify that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61201005), the Fundamental Research Funds for the Central Universities (No.ZYGX2011J006 and No.ZYGX2012J001) and the National Major Projects (2011ZX03003-003-04).

References

[1] A. S. Manne, "On the job-shop scheduling problem," *Operations Research*, vol. 8, no. 2, pp. 219–223, 1960.

[2] M. H. Rothkopf, "Scheduling with random service times," *Management Science*, vol. 12, no. 9, pp. 707–713, 1966.

[3] R. R. Muntz and E. G. Coffman Jr., "Preemptive scheduling of real-time tasks on multiprocessor systems," *Journal of the ACM*, vol. 17, no. 2, pp. 324–338, 1970.

[4] T. Malone, "Enterprise: a market-like task scheduler for distributed computing environments," *The Ecology of Computation*, pp. 177–205, 1988.

[5] X. He, X. Sun, and G. Von Laszewski, "QoS guided Min-Min heuristic for grid task scheduling," *Journal of Computer Science and Technology*, vol. 18, no. 4, pp. 442–451, 2003.

[6] D. Geer, "Industry trends: chip makers turn to multicore processors," *Computer*, vol. 38, no. 5, pp. 11–13, 2005.

[7] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proceedings of the 38th Design Automation Conference*, pp. 684–689, June 2001.

[8] W. Vereecken, W. Van Heddeghem, D. Colle, M. Pickavet, and P. Demeester, "Overall ICT footprint and green communication technologies," in *Proceedings of the 4th International Symposium on Communications, Control, and Signal Processing (ISCCSP '10)*, pp. 1–6, March 2010.

[9] J. Huang, C. Buckl, A. Raabe, and A. Knoll, "Energy-aware task allocation for network-on-chip based heterogeneous multiprocessor systems," in *Proceedings of the 19th International Euromicro Conference on Parallel, Distributed, and Network-Based Processing (PDP '11)*, pp. 447–454, February 2011.

[10] B. Ge, N. Jing, W. He, and Z. Mao, "Contention and energy aware mapping for real-time applications on Network-on-Chip," in *Proceedings of the International Conference in SoC Design Conference (ISOCC '12)*, pp. 72–76, 2012.

[11] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European Journal of Operational Research*, vol. 2, no. 6, pp. 429–444, 1978.

[12] A. Merkel, J. Stoess, and F. Bellosa, "Resource-conscious scheduling for energy efficiency on multicore processors," in *Proceedings of the 5th ACM EuroSys Conference on Computer Systems (EuroSys '10)*, pp. 153–166, April 2010.

[13] O. Sathappan, P. Chitra, P. Venkatesh, and M. Prabhu, "Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 146–158, 2011.

[14] J. M. N. Abad, S. K. Shekofteh, H. Tabatabaee, and M. Mehrnejad, "CoreIIScheduler: scheduling tasks in a multi-core-based grid using NSGA-II technique," in *Intelligent Informatics*, pp. 507–518, Springer, New York, NY, USA, 2013.

[15] H. F. Sheikh and I. Ahmad, "Fast algorithms for thermal constrained performance optimization in DAG scheduling on multi-core processors," in *Proceedings of the International Green Computing Conference (IGCC '11)*, pp. 1–6, July 2011.

[16] A. J. Ruiz-Torres and F. J. López, "Using the FDH formulation of DEA to evaluate a multi-criteria problem in parallel machine scheduling," *Computers and Industrial Engineering*, vol. 47, no. 2-3, pp. 107–121, 2004.

[17] F. Samman, T. Hollstein, and M. Glesner, "New theory for deadlock-free multicast routing in wormhole-switched virtual-channelless networks-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 544–557, 2011.

[18] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *Journal of Systems Architecture*, vol. 50, no. 2-3, pp. 105–128, 2004.

[19] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of the 39th Annual Design Automation Conference (DAC '02)*, pp. 524–529, June 2002.

[20] E. Carvalho, N. Calazans, and F. Moraes, "Heuristics for dynamic task mapping in NoC-based heterogeneous MPSoCs," in *Proceedings of the 18th IEEE/IFIP International Workshop on Rapid System Prototyping (RSP '07)*, pp. 34–40, May 2007.

[21] M. Zakarya, N. Dilawar, M. A. Khattak, and H. Maqssod, "Energy efficient workload balancing algorithm for real-time tasks over multi-core," *World Applied Sciences Journal*, vol. 22, no. 10, pp. 1431–1439, 2013.

[22] R. D. Banker, A. Charnes, and W. W. Cooper, "Some models for estimating technical and scale inefficiencies in data envelopment analysis," *Management Science*, vol. 30, no. 9, pp. 1078–1092, 1984.

[23] D. Deprins, L. Simar, and H. Tulkens, "Measuring labor-efficiency in post offices," in *Public Goods, Environmental Externalities and Fiscal Competition*, pp. 285–309, Springer, New York, NY, USA, 2006.

[24] J. Doyle and R. Green, "Efficiency and cross-efficiency in DEA: derivations, meanings and uses," *Journal of the Operational Research Society*, vol. 45, no. 5, pp. 567–578, 1994.

[25] D. McFadden, "Cost, revenue, and profit functions," in *Histoy of Economic Thought Chapters*, vol. 1, 1978.

[26] P. J. Agrell and J. Tind, "A dual approach to nonconvex frontier models," *Journal of Productivity Analysis*, vol. 16, no. 2, pp. 129–147, 2001.

[27] O. Sinnen, L. A. Sousa, and F. E. Sandnes, "Toward a realistic task scheduling model," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 3, pp. 263–275, 2006.

[28] A. Makhorin, "GLPK (GNU linear programming kit)," 2006.

[29] H. O. Fried, C. A. K. Lovell, and S. S. Schmidt, *The Measurement of Productive Efficiency and Productivity Growth*, Oxford University Press, Oxford, UK, 2008.

[30] R. P. Dick, D. L. Rhodes, and W. Wolf, "TGFF: task graphs for free," in *Proceedings of the 1998 6th International Workshop on Hardware/Software Codesign*, pp. 97–101, March 1998.

[31] M.-Y. Wu and D. D. Gajski, "Hypertool: a programming aid for message-passing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 3, pp. 330–343, 1990.

[32] S. Kim and J. Browne, "A general approach to mapping of parallel computation upon multiprocessor architectures," in *Proceedings of the International Conference on Parallel Processing*, 1988.

[33] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[34] C. Song, W. Chang, L. Yubai, and Y. Zhongming, "A NoC simulation and verification platform based on systemC," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 423–426, December 2008.

[35] A. B. Kahng, L. Bin, L. Peh, and K. Samadi, "Orion 2.0: a fast and accurate NoC power and area model for early-stage design space exploration," in *Proceedings of the conference on Design, Automation and Test in Europe (DATE '09)*, pp. 423–428, April 2009.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization