Industrial Engineering
& Management Systems
Vol 11, No 3, September 2012, pp.233-240
ISSN 1598-7248 | EISSN 2234-6473 |
http://dx.doi.org/10.7232/iems.2012.11.3.233
© 2012 KIIE

Generation of Business Process Reference Model Considering Multiple Objectives

Bernardo Nugroho Yahya
School of Technology Management, Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea

Jei-Zheng Wu
Department of Business Administration, Soochow University, Taipei, Taiwan

Hyerim Bae*
Department of Industrial Engineering, Pusan National University, Busan, Korea

(Received: February 4, 2012 / Revised: June 18, 2012 / Accepted: June 25, 2012)

ABSTRACT

The implementation of business process management (BPM) systems in large number of business organizations transforms BPM system into such a level of maturity and tends to collect large repositories of business process (BP) models. This issue encourages BP flexibility that leads to a large number of process variants derived from the same model, but differing in structure, to be stored in the large repositories of BP models. Therefore, the repositories may include thousands of activities and related business objects with variation of requirements and quality of service. It is a common practice to customize processes from reference processes or templates in order to reduce the time and effort required to design and deploy processes on all levels. In order to address redundancy and underutilization problems, a generic process model, called as reference BP, is absolutely necessary to cover the best of process variants. This study aims to develop multiple-objective business process genetic algorithm (MOBPGA) to find a set of non-dominated (Pareto) solutions of business reference model to enhance conventional approach which considered only a single objective on creating BP reference model by using proximity score measurement. A mixed-integer linear program is constructed to evaluate performance of the proposed MOBPGA on small-scale problems by using standard measures for multiple-objective techniques. The results will show the viability of applying MOBPGA in terms of simultaneously maximizing proximity score measurement, minimizing total duration, and total costs of the selected reference model.

Keywords: Business Process Management, Business Process Reference Model, Multi-Objective, Genetic Algorithm

* Corresponding Author, E-mail: hrbae@pusan.ac.kr

1. INTRODUCTION

It is a common practice to customize processes from reference processes or templates in order to reduce the time and effort required to design and deploy processes on all levels (Lu and Sadiq, 2006). In order to address redundancy and underutilization problems, a generic process model, called as business process (BP) reference, is absolutely necessary to cover the best of process variants. Industrial process model collections and reference process models (e.g., ITIL, SCOR, eTOM) are some related concepts, practices and standards for developing such a process model. However, those concepts are settled as a high-level model without considering the customization implementation. Customization is a major trend that occurs in many industries and markets resulting from the customer's specific demand on goods and services.

The notion of BP variant is important in explaining the customization in this study. We use the terminology of BP variant to represent the different kind of customized BP with almost similar business objectives. For example, a supplier offers, specifies, produces and delivers different kinds of products to its customers, or suppliers have different kinds of actor relationships that separate orders and frame contracting. Due to the different kinds of BP, there will be a problem in resolving which process is appropriate to a given organization. The difficulty derives from the fact that the activities involved in taking an account to contribute in two or more than hundreds of process. It is certain that a "good" process should be obtained. In order to obtain a "good" process, it is necessary to find a generic process, later called as reference model, that considers a part of sequential view of process with an understanding of the business properties included in the activity.

Approaches to the generation of BP reference models have been explored. Previous work on BP reference model generation has utilized proximity score measurement (PSM) and dealt with activity proximity and frequency as well as validity (Yahya *et al.*, 2010). A reference model with only a single alternative solution is equally unsatisfactory to stakeholder. Cost and duration, indeed, are two additional important factors relevant to business objectives. In contrast to the single-objective case, a multi-objective approach can be used to simultaneously optimize, subject to certain constraints, two or more conflicting objectives so as to find a valid and representative BP reference model. Multi-objective optimization is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. By using multi-objective optimization, it may bring some advantages in finding a valid and representative BP reference models compared with single objective problem.

This study presents a method of solving the multi-objective optimization problem by using forms of genetic algorithms. In multi-objective genetic algorithms we attempt to optimize not only one fitness parameter but also a collection of them. To achieve this, we generate a method of measuring the overall fitness of the set of objectives.

This paper is organized as follows. Section 2 discusses works related to this study. The proposed approach is described in Section 3. The result of experiments is presented in Section 4 and Section 5 concludes this study.

2. LITERATURE REVIEW

Many efforts have been made on BP modeling research and research on finding reference BP based on BP models is on the increase. van der Aalst *et al.* (2005) discussed configurable process models as a basis for reference modeling. Li *et al.* (2009) extended previous research by proposing a heuristic approach to discovering reference models using a measurement to analyze the distance of change operations in modeling. It updates process configurations and produces new reference models based on a minimum edit distance from initial reference processes. A mathematical programming approach to discovering new reference models was introduced to address the issue of creating reference processes without initial reference information or process reconfiguration. There remain problems in the presentational and validation aspects of the process using interger programming formulations, which this study attempttted to overcome with a genetic algorithm approach in (Yahya *et al.*, 2010).

Multi-objective genetic algorithms (MOGA) have been widely used in many areas including order allocation (Wu *et al.*, 2012), scheduling problem (Zinflou *et al.,* 2008; Minella *et al.*, 2008), human resource planning (Abboud *et al.*, 1998; Hajri-Gabouj, 2003) or design and planning problem (Zhou *et al.*, 2003; Gen *et al.*, 2009). Deb *et al.* (2002) used genetic algorithm to solve multi-objective problems and develop non-dominated sorting and further brought up the idea of using the elitist strategy may have a better convergence when using the genetic algorithm (GA). Research using GA on graph topology is limited to supply chain network (Wang *et al.*, 2010). Hence, this study proposes a novel approach on BP modeling domain.

In the domain of BP, Quan and Tian (2009) presented a research on business processes' multi-objective optimization to point out at three criteria of variables: processing time, process cost, and quality. The proposed research used simulation as the result experiments. Vergidis *et al.* (2006, 2007) carried out researches on BP improvement using multi-objective optimization and composite BP using an approach of evolutionary multi-objective optimization approach. Neubauer and Heurix (2008a, 2008b) discussed a case study a multi-objective decision support for defining secure BP. However, none of those studies point out variables related to the process structure behavior which is one of the important aspects on generating reference model.

3. PROPOSED APPROACH

3.1 Mixed-Integer Linear Programming Model

This section provides definitions of process model and structural schema to describe a formal model of BP.

Definition 1 (Process model).
We define a process model p^k as the k-th process in a process repository. It can be represented as a tuple of $<A^k, L^k>$, each element of which is defined below.

- $A^k = \{a_i | \ i = 1, \ \cdots, I\}$ is a set of activities where a_i is the i-th activity of p^k and I is the total number of activities in p^k.

- **A** is defined as a set of all activities in the process repository, where A is the union of all A^k, $\mathbf{A} = \bigcup_{k=1}^{K} A^k$
- $L^k \subseteq \{l_{ij} = (a_i, a_j) \mid a_i, a_j \in A^k\}$ is a set of links where l_{ij} is the link between two activities a_i and a_j in the k-th process. The element (a_i, a_j) represents the fact that a_i immediately precedes a_j.
- a_{i+} is the activity following a_i and a_{i}- is the activity preceding a_i
- **L** is a set of all links in the process repository, where **L** is the union of all L^k, $k = 1, \cdots, K$, i.e., $\mathbf{L} = \bigcup_{k=1}^{K} \mathbf{L}^k$
- For a split activity a_i, such that $|N_i| > 1$, where $N_i = \{a_{j+} |(a_i,a_{i+}) \in L\}$, $f(a_{i+})$ = 'AND' if all a_{i+} should be executed; otherwise, $f(a_{i+})$ = 'XOR.'
- For a merge activity a_i, such that $|M_i| > 1$, where $M_i = \{a_{j-}|(a_{i-}, a_i) \in L\}$, $f(a_{i-})$ = 'AND' if all a_{i-} should be executed; otherwise, $f(a_{i-})$ = 'XOR.'
- In order to collect information on the variety of start and end activities among process variants, we define $\boldsymbol{A_S}$ and $\boldsymbol{A_E}$ as the sets of start and end activities, respectively.
- $\boldsymbol{A_S} = \{a_i| a_i \in \boldsymbol{A^k}, |M_i| = 0, \forall k \in K\}$ is a set of start activities in all K process variants where a_i is an activity in p^k with an empty set of preceding activities, $|M_i| = 0$.
- $\boldsymbol{A_E} = \{a_i| a_i \in \boldsymbol{A^k}, |N_i| = 0, \forall k \in K\}$ is a set of end activities in all K process variants where a_i is an activity in p^k with an empty set of succeeding activities, $|N_i| = 0$.

Definition 2 (Structural flow schema).
A structural schema is a set of tuples, each of which has an index and link information. A structural schema (**H**) required for the chromosome generation is defined as

$$\mathbf{H} = \{(s, l_{ij}) \mid s = 1, 2, \cdots, S, \text{ and } l_{ij} \in \mathbf{L}\}$$

The value of s will start from 1 and increase accordingly until all links are assigned. The value of s increases in different ways according to control flow types, sequential flow and split-merge flow. For each l_{ij} in a sequential flow, increasing number of s will be assigned, and each l_{ij} has a unique integer value of s. For all l_{ij}'s from a split activity or to a merge activity, a single value of s will be assigned.

Index and subscript

K : the number of process variants
K : process variant index, $k = 1, \cdots, K$
S : structural flow schema index, $s = 1, \cdots, S$
i, j, r, t : activity index

Set

p^k : process k in a process repository
$\boldsymbol{A^k}$: a set of activities in process k
$\boldsymbol{L^k}$: a set of links in process k
A : a set of all activities in the process repository
$\boldsymbol{A_S}$: a set of start activities among K processes
$\boldsymbol{A_E}$: a set of end activities among K processes
L : a set of all links in the process repository
H : a set of structural flow schema in a group of process variants
$\mathbf{P}_i$: a set of immediately preceding activities (predecessors) of activity i in the process repository
$\mathbf{S}_i$: a set of immediately succeeding activities (successors) of activity i in the process repository

Parameter

C_i : expected cost to execute i-th activity
C_{ij} : proximity score for executing a link (a_i, a_j) ($\in$ **L**)
E_i : expected time to execute i-th activity
N_i : number of predecessors of i-th activity
M_i : number of successors of i-th activity
V_s : coordination time between activities in schemes.
W_s : average coordination time. $W_s = V_s / \sum_{(s,\, l_{ij}) \in \mathbf{H}} 1$ (derived from average duration divided by number of links in one structural scheme)
M : big M, a very large positive
r_i : ready time for i-th activity
f_i : finish time for i-th activity. Particularly, $f_i = r_i + E_i$

Decision variable

v_i : equals to 1 if activity is selected, $a_i \in$ **A**; 0, otherwise
y_i : equals to 1 if activity $a_i \in \boldsymbol{A_S}$ is a start activity and an element of the set of start activities; 0, otherwise
z_i : equals to 1 if activity $a_i \in \boldsymbol{A_E}$ is an end activity and an element of the set of end activities; 0, otherwise
x_{ij}^s : equals to 1 if activity a_i immediately precedes activity a_j in scheme s, $(a_i, a_j) \in$ **L**, $(s, l_{ij}) \in$ **H**; 0, otherwise

$$\min \sum_{(a_i, a_j) \in \mathbf{L}} \left[(K - C_{ij}) \sum_{(s, l_{ij}) \in \mathbf{H}} x_{ij}^s + C_{ij} (1 - \sum_{(s, l_{ij}) \in \mathbf{H}} x_{ij}^s) \right] \quad (1)$$

$$\min \ C_{\max} \quad (2)$$

$$\min \sum_{i (a_i \in \mathrm{A})} C_i v_i \quad (3)$$

Note that a tri-objective integer programming model for generating a BP reference model is provided as an extension of a single objective problem. First, it attempts to find the maximum proximity score, (notice that it has been converted into a minimization problem), among the links existing in the process variants. The BP reference should satisfy the business requirements, which are defined as duration (2) and cost (3). Activity duration, called makespan, is a measure to evaluate the expected execution time of a BP reference model. In respect to cost, it will sum the costs of all selected activities for the best BP reference model.

In particular, the constraint can be decomposed into four segments; scheme (4)-(5), predecessors (6)-(9), successors (10)-(13), and predefined duration (14)-(16), respectively. Scheme constraint (4) specifies that a scheme behavioral relation of an activity should be less than or equal to 1. Among all structural schema relations, it should be a scheme that is consistent with (5). With regard to predecessors, constraint (6) expresses that there should be at least one link entering activity i if activity i exists and activity i is not a start activity. To find a start activity, there should be a constraint to certify that there are no immediate predecessors to activity i (7). Certainly, activity consistency constraint (8) is necessary, that is, that an (start) activity is selected in the BP reference. In addition, we apply constraint (9) to specify that only an activity from a set of start activities (A_S) can be a start activity, thereby restricting others (10).

Constraints

$$\sum_{s,i,j} x_{ij}^s \le 1,\ \forall (s,\ l_{ij}) \in \mathbf{H} \tag{4}$$

$$x_{ij}^s = x_{rt}^s,\ \forall s,\ (s,\ l_{ij}),\ (s,\ l_{rt}) \in \mathbf{H} \tag{5}$$

$$y_i + \sum_{s,j,i} x_{ji}^s \ge v_i,\ \forall a_i \in \mathbf{A},\ (s,\ l_{ji}) \in \mathbf{H},\ N_i > 0 \tag{6}$$

$$N_i y_i + \sum_{s,j,i} x_{ji}^s \le N_i v_i,\ \forall a_i \in \mathbf{A},\ (s,\ l_{ji}) \in \mathbf{H},\ N_i > 0 \tag{7}$$

$$y_i = v_i,\ \forall a_i \in \mathbf{A},\ N_i = 0 \tag{8}$$

$$\sum_i y_i = 1,\ \forall a_i \in A_S \tag{9}$$

$$\sum_i y_i = 0,\ \forall a_i \notin A_S \tag{10}$$

$$z_i + \sum_{s,i,j} x_{ij}^s \ge v_i,\ \forall a_i \in \mathbf{A},\ (s,\ l_{ij}) \in \mathbf{H},\ M_i > 0 \tag{11}$$

$$M_i z_i + \sum_{s,i,j} x_{ij}^s \le M_i v_i,\ \forall a_i \in \mathbf{A},\ (s,\ l_{ij}) \in \mathbf{H},\ M_i > 0 \tag{12}$$

$$z_i = v_i,\ \forall a_i \in \mathbf{A},\ M_i = 0 \tag{13}$$

$$\sum_i z_i = 1,\ \forall a_i \in A_E \tag{14}$$

$$\sum_i z_i = 0,\ \forall a_i \notin A_E \tag{15}$$

$$\mathrm{M}(\sum_{s,i,j} x_{ij}^s - 1) + f_i + \sum_s W_s \sum_{s,r,t} x_{rt}^s \le r_j,\ \forall (a_i,\ a_j) \in \mathbf{L}, (s,\ l_{ij}),\ (s,\ l_{rt}) \in \mathbf{H} \tag{16}$$

$$r_i + E_i = f_i,\ \forall a_i \in \mathbf{A} \tag{17}$$

$$f_i \le C_{\max},\ \forall a_i \in \mathbf{A} \tag{18}$$

$$x_{ij}^s \in \{0,\ 1\},\ \forall (s,\ l_{ij}) \in \mathbf{H} \tag{19}$$

$$v_i,\ y_i,\ z_i \in \{0,\ 1\},\ \forall a_i \in \mathbf{A} \tag{20}$$

Subsequently, with regard to successors, constraint (11) expresses that there should be at least a link leaving activity i if activity i exists and activity i is not an end activity. To find an end activity, there should be a constraint to certify that there are no immediate successors to activity i (12). Certainly, we need to apply activity consistency constraint (13) to ensure that an (end) activity is selected in the BP reference. In addition, we set constraint (14) to specify that an activity can be an end activity only if it is from a set of end activities (A_E), and restrict, thereby, others (15).

Constraints (16)-(19) relate to process duration. Constraint (16) denotes that the ready time of activity j should be greater or the same as the finish time of activity I, including the coordination time and the average duration that may occur during split or merge. Constraint (17) defines the relations among ready time, execution time and finish time. Constraint (18) specifies the makespan. Constraint (19) ensures that the adjacent relationship of activities i and j includes a binary value element. Constraint (20) specifies that all activities (including start and end activities) has a binary value element.

3.2 Multiple-Objective BP Genetic Algorithm

This section proposes a solution to the BP reference selection problem, using a multi-objective genetic algorithm with maximizing proximity score measurement, cost minimization and duration minimization objectives (MOBPGA). MOBPGA considers four parameters, including generation size (*gen_size*), population size (*pop_size*), crossover rate (r_1), and mutation rate (r_2) (Gen and Cheng, 2000). The generation, denoted by t, represents the number of computation iterations of the GA. It contains a set of solutions that collectively corresponds to a population, denoted as $P(t)$. Chromosomes, which represent the solution, consist of genes that encode genotypes for evolution, and should be evaluated based on specific measures of fitness during each generation. In order to generate new offspring $O(t)$, a new generation of chromosomes, MOBPGA employs crossover and mutation operations.

Procedure: Multi-objective business process genetic algorithm (MOBPGA)

```
begin
    initialize P(t);
    evaluate P(t) based on the NSGAII;
    generate Pareto solutions;
    for t = 1 to gen_size do
        recombine P(t) to yield O(t) based on the partial-mapped crossover and random key representation mutation;
        evaluate O(t) based on the NSGAII;
        update Pareto solutions;
        select P(t+1) from P(t) and O(t) based on the composite Pareto ranking selection (cPrs);
    end
end
```

The MOBPGA begins with initialization and ends when the number of run generations reaches the generation size. Following several generations of evolution, the algorithms improve solution quality and, in some cases, converge to the optimal solutions of the problem. Increasing the population size, generation size, crossover rate, or mutation rate explores more of the solution space while requiring greater computational effort. The final decision is made by the decision makers, who manually select the preferred Pareto solution according to their preference structures.

The proposed MOBPGA includes specific random key representation for encoding and decoding, partially mapped crossover, random key representation mutation, an algorithm for efficient Pareto sorting (NSGAII), and composite Pareto ranking selection (cPrs) (Goldberg, 1989). Figure 1 shows the random key representation of encoding method used in this study.

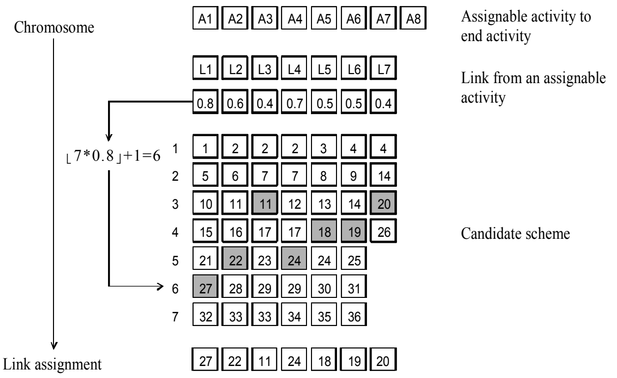

Figure 1. Random key representation encoding method.

Due to the random genetic processes (crossover and mutation) that might produce an invalid result, it is necessary to verify the process for further generation. We apply the soundness properties of business processes to verify the process well-formedness. For a business process model to be sound, three properties are required (van der Aalst, 2000): 1) for every activity a_j that is reachable from a_i, there exists a sequence to the next activity until a preferred end activity a_E is reached, 2) a preferred end activity a_E is the activity that should be reachable from a start activity a_S, and 3) there is no activity that is never processed in any execution of the model (there are no needless elements).

In order to generate a valid BP, there should be a decoding mechanism that follows some procedures. In this approach, there are three important mechanisms of decoding mechanism for establishing a valid BP. First, it checks the *start* and *end* activity. Second, it should check the *scheme* (mapping processes and links). Finally, it will check the path which contains candidate links. There are four steps to execute the decoding mechanism which is as follow.

Step 1: Disqualify schemes flowing to *Start* of flowing from *End*.
Step 2: Disqualify pair of activities that are not existed in the repository
Step 3: Disqualify activities that are not available in the path
Step 4: Restructure feasible activities and schemes

4. EXPERIMENT RESULTS

In order to validate our approach, several experiments were conducted. The experiment estimates the validity of the proposed algorithm in real settings by using logistics processes that contain a various types of process structure. Numerical analysis was performed on a desktop computer equipped with an Intel® Pentium® D 3.40GHz CPU and 2GB RAM. The commercial software LINGO 11.0 (LINGO System) was used to generate a reference set of non-dominated solutions, **R**, utilizing embedded integer programming (IP) packages. For each scenario, LINGO solved the weighted-sums problem with objectives (1)-(3) subject to (4)-(20), which was recognized as ILP (shown in Lingo) and terminated at local optimal solutions. For each problem instance, a local optimal solution was collected within various ranges of computation time. Hence, each scenario has a set of non-dominated solutions for further comparison. The performance metrics discussed above were used to compare the performances of MOBPGA and multi-objective mixed integer linear programming (MoMILP).

We conducted an experiment with various numbers of activities among four process groups (PG). Four groups from PG-I to PG-IV, each consisting of ten process variants (p_1–p_{10}), were formed. Process variants are generated from a single process model, each of which fits a certain scenario and context; in other words, the configuration of a particular process variant reflects the specific requirement and circumstance of the process. The gradual increase in the number of activities of each process group is shown in Table 1. Since there is no exact cost and duration time data on our logistics process models, we express them using random integer values representing units of metric rather than real numbers for any currency or time.

Table 1. Mean number of activities in each group of 4 processes

Process group	PG-I	PG-II	PG-III	PG-IV
Mean no. of activities	6.1	10.4	14.8	19
Max no. of activities	7	11	15	20

Based on our results using process group III (PG-III), the single-objective approach can produce only a single process reference model, with no guarantee of meeting some requirements (Figure 2). Our proposed approach has several alternatives for decision makers. For example, a process manager might want to select a BP reference model for duration of less than 30 units of time. Some of these alternative candidates are shown in Figure 3. The choice of generating given alternatives will correspond to the stakeholders' business requirements and will accrue benefits to business organizations in terms of strategic competitiveness.

When we applied process variants of activity number 20 (PG-IV), MoMILP computation time was intractable. Thus, we set a cut-off time of 21,600 seconds to obtain feasible solutions. The MOBPGA parameters operative were 1,000 generations, a population size of 20, a mutation rate of 0.8 and multiplier (set on 0.9) for the purpose of selection. Experiment result is shown in Table 2. The computational time of the approaches,

Table 2. Experiment result

	MoMILP			MOBPGA			Reference Pareto		
	PSM	Cost	Time	PSM	Cost	Time	PSM	Cost	Time
PG-I	46	22	35				46	22	35
PG-I				47	34	46			
PG-I	48	12	22				48	12	22
PG-I				51	12	24			
PG-I	54	10	21				54	10	21
PG-I	56	12	19				56	12	19
PG-I				57	21	13	57	21	13
PG-I				57	10	21			
PG-II	61	36	35	61	36	35	61	36	35
PG-II	65	33	31	65	33	31	65	33	31
PG-II	65	36	29				65	36	29
PG-II	69	33	25				69	33	25
PG-II				75	48	27			
PG-II				79	45	23	79	45	23
PG-II	81	33	22				81	33	22
PG-II	81	28	54				81	28	54
PG-II	107	36	19				107	36	19
PG-II				117	72	22			
PG-III				114	87	54	114	87	54
PG-III				116	99	42	116	99	42
PG-III				116	83	44	116	83	44
PG-III				116	78	46	116	78	46
PG-III				116	69	52	116	69	52
PG-III				118	78	35	118	78	35
PG-III				118	74	47	118	74	47
PG-III				118	65	55	118	65	55
PG-III				120	69	30	120	69	30
PG-III				120	60	35	120	60	35
PG-III				122	68	32	122	68	32
PG-III				122	59	39	122	59	39
PG-III	122	54	41				122	54	41
PG-III				124	74	27	124	74	27
PG-III	124	54	28				124	54	28
PG-III				126	89	23	126	89	23
PG-III				126	69	28			
PG-III				126	56	30			
PG-III	126	54	25				126	54	25
PG-III	126	45	29				126	45	29
PG-III				128	55	40			
PG-III	128	45	23				128	45	23
PG-III				130	74	24			
PG-III				132	67	25			
PG-III				132	51	36			
PG-III				132	48	61			
PG-III				134	65	26			
PG-III				134	50	43			
PG-III				134	44	49			
PG-III	134	41	20						
PG-III				136	59	28			
PG-III				136	55	35			
PG-III				136	39	37	136	39	37
PG-III				138	63	24			
PG-III				138	62	26			
PG-III				138	54	29			
PG-III				138	41	31			
PG-III				138	38	49			
PG-III	138	35	30				138	35	30
PG-III				140	108	20	140	108	20
PG-III				140	57	27	140	57	27

	MoMILP			MOBPGA			Reference Pareto		
	PSM	Cost	Time	PSM	Cost	Time	PSM	Cost	Time
PG-III	140	35	24				140	35	24
PG-III				142	59	21			
PG-III				142	35	34			
PG-III				144	58	23			
PG-III				144	50	26			
PG-III				144	37	28			
PG-III				144	34	40	144	34	40
PG-III	144	37	18				144	37	18
PG-III				146	53	24			
PG-III				148	55	18			
PG-III				148	48	25			
PG-III				150	46	23			
PG-III				154	42	26			
PG-III				154	33	33	154	33	33
PG-III				156	44	20			
PG-III				156	35	27			
PG-III	158	37	15				158	37	15
PG-III	158	29	30				158	29	30
PG-III				160	38	23			
PG-III				160	29	30			
PG-III	160	29	23				160	29	23
PG-III				162	40	17			
PG-III				162	31	24			
PG-III	164	31	17				164	31	17
PG-IV				158	58	99	158	58	99
PG-IV				162	76	88	162	76	88
PG-IV				166	71	83	166	71	83
PG-IV				168	67	87	168	67	87
PG-IV				168	56	102	168	56	102
PG-IV				170	68	83	170	68	83
PG-IV				170	56	94	170	56	94
PG-IV				172	52	93	172	52	93
PG-IV	172	52	106						
PG-IV				174	106	77	174	106	77
PG-IV				174	76	79	174	76	79
PG-IV				176	91	77	176	91	77
PG-IV				180	67	81	180	67	81
PG-IV				180	56	92	180	56	92
PG-IV				182	84	73	182	84	73
PG-IV				182	61	82	182	61	82
PG-IV				182	50	83	182	50	83
PG-IV				184	83	72	184	83	72
PG-IV				184	62	78	184	62	78
PG-IV				186	104	66	186	104	66
PG-IV				186	79	76	186	79	76
PG-IV				186	68	77	186	68	77
PG-IV				188	68	66			
PG-IV	188	50	58				188	50	58
PG-IV				192	59	80			
PG-IV				200	98	65			
PG-IV	200	50	50				200	50	50
PG-IV				204	65	77			
PG-IV				204	53	80			
PG-IV				206	55	79			
PG-IV	206	56	45				206	56	45
PG-IV	210	56	45						
PG-IV				220	101	62	220	101	62
PG-IV	226	47	99				226	47	99

PG: process group, PSM: proximity score measurement, MoMILP: multi-objective mixed integer linear programming, MOBPGA: multi-objective business process genetic algorithm.

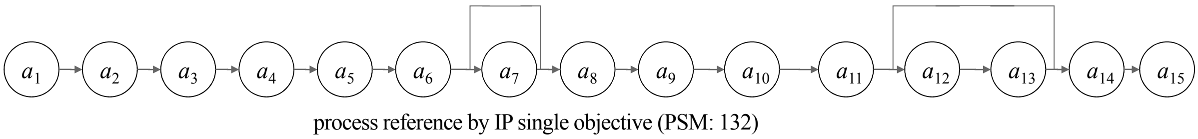

Figure 2. Generated process reference model using genetic algorithm approach (Yahya *et al*., 2010).
PSM: proximity score measurement.

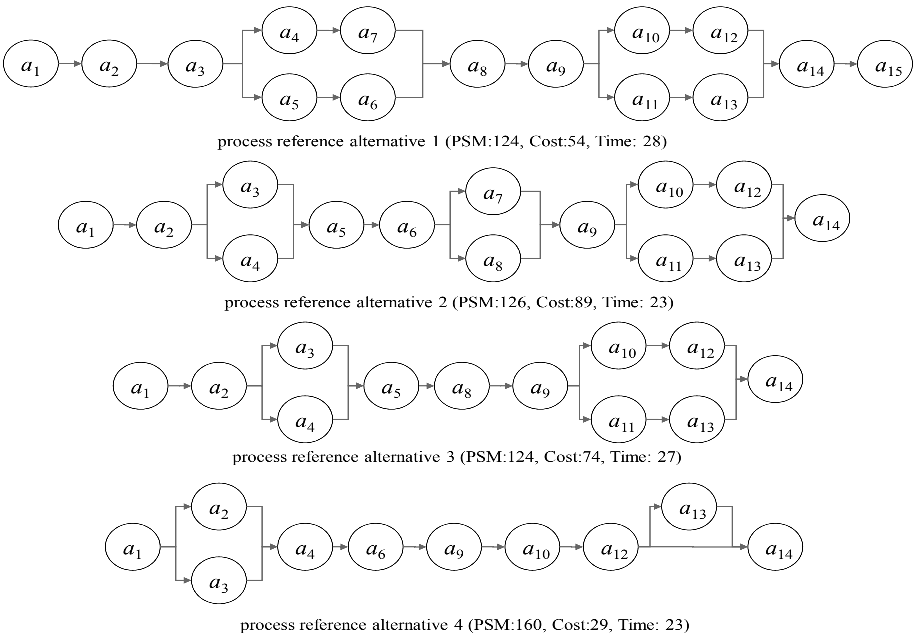

Figure 3. Some alternatives of process reference considering time <30.
PSM: proximity score measurement

MoMILP and MOBPGA, are listed in Table 3. MoMILP sets a cut-off time of 21,600 seconds for feasible solutions, compared with MOBPGA's 222 seconds for an activity num-ber of 20. Finally, the performance of the MOBPGA is robust and our performance evaluations showed little dif-ference from MoMILP in terms of quality of the solutions.

Table 3. Performance evaluations of three scenarios using MoMILP and MOBPGA

Scenario	Max no. activities	Method	*RT* (sec)
PG-I	7	MoMILP	6.0
		MOBPGA	3.7
PG-II	11	MoMILP	107.0
		MOBPGA	13.8
PG-III	15	MoMILP	2,674.0
		MOBPGA	140.1
PG-IV	20	MoMILP	21,600.0
		MOBPGA	222.0

PG: process group, MoMILP: multi-objective mixed integer linear programming, MOBPGA: multi-objective business process genetic algorithm.

5. CONCLUSION

This paper proposes a MOBPGA with specific random key representation of encoding and decoding to solve the problem of BP reference model generation with objective functions maximizing process proximity while minimizing cost and duration. The proposed model meets customer requirements by coordinating objectives, specifically by incorporating the process topology into the operational objectives. It also presents a compromise approach that can facilitate decision making in the context of a large set of Pareto solutions, thereby enabling decision makers to incorporate other preferences or business concerns. By utilizing the results of non-dominated solutions both from MOBPGA and MoMILP, we can suggest alternative solutions to stakeholders based on certain conditions. Compared with the conventional ap-proach, this provides more solutions to decision makers, enabling them to select process reference models based on their requirements.

BP reference model generation will have to be improved, considering the following limitations. First, this study employed experiments using the block concept of process structure. Whereas there are some benefits on the experiments, this approach covers a little issue with regard to deadlock and flawless process in BP. Moreover, there are some issues remaining with regard to operation dependency within activities. Second, since the graph generation was assumed to be flawless, there is a need to carefully consider the control flow semantics such as AND or OR control-flow in generating the BP reference model. These are issues that will require further research.

ACKNOWLEDGMENTS

This work was partially supported by the National Research Foundation of Korea Grant funded by the Korea Government (No.2012R1A2008335) and by National Science Council, Taiwan (NSC100-2410-H-031-011-MY2).

REFERENCES

Abboud, N., Inuiguchi, M., Sakawa, M., and Uemura, Y. (1998), Manpower allocation using genetic annealing, *European Journal of Operational Research*, **111**(2), 405-420.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, **6**(2), 182-197.

Goldberg, D. E. (1989), *Genetic Algorithms in Search*, *Optimization*, *and Machine Learning*, Addison-Wesley, Reading, MA.

Gen, M. and Cheng, R. (2000), *Genetic Algorithms and Engineering Optimization*, John Wiley and Sons, New York, NY.

Gen, M., Katai, O., McKay, B., Namatame, A., Sarker, R. A., and Zhang, B.-T. (2009), *Intelligent and Evolutionary Systems*, Springer-Verlag, Berlin, Germany.

Hajri-Gabouj, S. (2003), A fuzzy genetic multiobjective optimization algorithm for a multilevel generalized assignment problem, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part C: Applications and Reviews*, **33**(2), 214-224.

Li, C., Reichert, M., and Wombacher, A. (2009), Discovering reference models by mining process variants using a heuristic approach, *Business Process Management*, *Lecture Notes in Computer Science*, **5701**, 344-362.

Lu, R. and Sadiq, S. (2006), Managing process variants as an information resource, *Business Process Management*, *Lecture Notes in Computer Science*, **4102**, 426-431.

Minella, G., Ruiz, R., and Ciavotta, M. (2008), A review and evaluation of multiobjective algorithms for the flowshop scheduling problem, *INFORMS Journal on Computing*, **20**(3), 451-471.

Neubauer, T. and Heurix, J. (2008a), Defining secure business processes with respect to multiple objectives, *Proceedings of the 3rd International Conference on Availability*, *Reliability and Security*, Barcelona, Spain, 187-194.

Neubauer, T. and Heurix, J. (2008b), Multiobjective decision support for defining secure business processes: a case study, *International Journal of Business Intelligence and Data Mining*, **3**(2), 177-195.

Quan, L. and Tian, G.-S. (2009), A business processes' multi-objective optimization model based on simulation, *Proceedings of International Conference on Information Management*, *Innovation Management and Industrial Engineering*, Xi'an, China, 572-575.

Tiwari, A., Vergidis, K., and Turner, C. (2010), Evolutionary multi-objective optimization of business processes, *Soft Computing in Industrial Application*, *Advances in Intelligent and Soft Computing*, **75**, 293-301.

Van der Aalst, W. M. P. (2000), Workflow verification: finding control-flow errors using Petri-net-based techniques, *Business Process Management*, *Lecture Notes in Computer Science*, **1806**, 161-183.

Van der Aalst, W. M. P., Alves de Medeiros, A. K., and Weijters, A. J. M. M. (2005), Genetic process mining, *Applications and Theory of Petri Nets, Lecture Notes in Computer Science*, **3536**, 48-69.

Vergidis, K., Tiwari, A., and Majeed, B. (2006), Business process improvement using multi-objective optimization, *BT Technology Journal*, **24(2)**, 229-235.

Verdigis, K., Tiwari, A., Majeed, B., and Roy, R. (2007), Optimisation of business process designs: an algorithmic approach with multiple objectives, *International Journal of Production Economics*, **109**(1/2), 105-121.

Wang, L.-C., Cheng, C.-Y., and Huang, L.-P. (2010), A genetic algorithm for directed graph-based supply network planning in memory module industry, *Industrial Engineering and Management Systems*, **9**(3), 227-241.

Wu, J.-Z., Chien, C.-F., and Gen, M. (2012), Coordinating strategic outsourcing decisions for semiconductor assembly using a bi-objective genetic algorithm, *International Journal of Production Research*, **50**(1), 235-260.

Yahya, B. N., Bae, H., Bae, J., Kim, D. (2010) Generating business process reference model using genetic algorithm, *Proceedings of the 23th Annual Conference of Biomedical Fuzzy Association*, *Soft Computing Application*, Kitakyushu, Japan, 245-248.

Zhou, G., Min, H., and Gen, M. (2003), A genetic algorithm approach to the bi-criteria allocation of customers to warehouses, *International Journal of Production Economics*, **86**(1), 35-45.

Zinflou, A., Gagne, C., Gravel, M., and Price, W. L. (2008), Pareto memetic algorithm for multiple objective optimization with an industrial application, *Journal of Heuristics*, **14**(4), 313-333.