An Adaptive Approach for Solving Dynamic Scheduling with Time-varying Number of Tasks – Part I

Manuel Blanco ABELLO*, Lam Thu BUI†, Zbigniew MICHALEWICZ‡
*School of Computer Science, University of Adelaide, Adelaide, South Australia
Email: manuel.abello@adelaide.edu.au
†Le Quy Don Technical University, Viet Nam
Email: lam.bui07@gmail.com
‡School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia;
also at the Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland;
and at the Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland,
Email: zbyszek@cs.adelaide.edu.au

***Abstract*—Changes in environment is common in daily activities and usually introduce new problems. To be adaptive to these changes, new solutions to the problems are to be found every time change occur.**

Our previous publication showed that centroid of non-dominated solutions associated with Multi-Objective Evolutionary Algorithm (MOEA) from previous changes enhances the search quality of solutions for the current change. However, the number of tasks in the test environment employed was fixed. In this two-part paper, we address the dynamic adaptation with time-varying task number.

To cope with this variability, new components of the solution, corresponding to the new tasks, are inserted appropriately to all solutions of the previous changes. Then centroid of these modified solutions is recomputed. Further, to avoid confusion in solution presentation, the insertion of new tasks obliged the use of task ID number greater than the largest of the previous IDs. The first part of this paper will show that the resulting task numbering system will alter the centroid significantly which will degrade MOEA's search quality. To circumvent, task IDs are mapped to new values in order to minimize difference in IDs between adjacent solution components; an approach which significantly upgraded the search performance despite changes in task number as supported by the obtained results.

I. INTRODUCTION

Real-world problems often contain many uncertain and dynamic factors; i.e., air traffic scheduling is usually affected by unexpected events such as bad weather or emergencies. Therefore, it is unlikely that any solution found for these problems would stay valid for a long time. These factors require an adaptive mechanism to introduce changes to existing solution. Thus, new adaptive solution should be computed for each environmental change. This paper is the first part of a two-part paper dealing on the adaptation with number of task as a particular type of environmental change of interest. First and second parts will be called parts I and II, respectively.

The approach taken in our previous publication [1] to search for these solutions is Multi-Objective Evolutionary Algorithm (MOEA) [2]. It is possible that the adaptive solution for the current change is strongly related to solutions of some past changes, such that it should be worthwhile to use the latter as initial population for MOEA in order to speed up the adaptation process, i.e., the search of adaptive solution. In the said publication, a centroid is calculated for each set of non-dominated solutions obtained before a current change. It shows the overall tendency of its corresponding set towards the set of current adaptive solutions. These centroids become a part of MOEA's initial population to compute for the current adaptive solution. This technique is called Centroid-Based Adaptation (CBA).

Scheduling problems in situations with scarce resources is usually referred to as Resource-Constrained Project Scheduling (RCPS) [3]. Our publication in [1] dealt with RCPS whereby a solution is a string of scheduled tasks, labeled by ID numbers. It dealt on changes in task duration, task precedence, and availability of resources. Further, it sets the number of genotypes in a solution to be equal to the number of tasks. Parts I and II extend the types of changes to include the change in the number of tasks. With this extension, previous solutions (including the centroid) become inappropriate as initial individuals to MOEA when the current change is on the number of tasks. To circumvent this problem, new genes, which correspond to the new tasks, are inserted beside the genes which correspond to their immediate predecessors. This insertion is applied to all sets of non-dominated solution of previous changes and then centroid is recomputed for each of these sets.

To avoid confusion in RCPS' schedule presentation, the insertion of new tasks must use task ID numbers greater than the largest of the previous task IDs. Part I will show that the resulting task numbering system will degrade the optimization potential of centroid significantly. To circumvent this, task IDs are mapped in a manner that avoid large increment of IDs between adjacent genes. The approach implementing these innovations to CBA is called, Mapping of Task-IDs for

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

Centroid-Based Adaptation (McBA). Part II will relate the degree of change in environment to the performance of McBA.

Adaptive solution being sought for current change could have components already executed or are in progress; a type of problem called Dynamic Optimization Problems (DOPs) [4], [5]. Parts I and II choose the *Reactive* approach [6] to DOP whereby a pre-optimized solution is used as a baseline for scheduling. This baseline solution is revised or repaired to adapt to current changes in environment. The revision might incur a high cost or might be infeasible. Further, it must not be extreme to meet the primary objectives of the problem while at the same time minimize its cost; a trade-off that demands multi-objectivity [7], hence the use of MOEA. As CBA and McBA utilize past solutions for MOEA, they are classified as Memory-based approaches [8], [9]. None of the current literatures in these approaches employed solution representative such as in CBA or McBA. We choose RCPS due to its popularity in the study of adaptation in dynamic environments [3] and applied it in [1] to planning problem where it is renamed as Adaptive Planning Problem (APP).

Genes are inserted/removed to/from chromosomes for variable number of tasks in [10]. Tasks arrive randomly and put on queue then processed by batch where number of tasks in a batch varies in time, and so as chromosome length, in [11]. Ingenious technique to insert tasks was employed in [12] which minimizes deviation from makespan and at the same time keep vital properties of baseline schedule. More techniques on coping with task number variation can be found in [13]. None of current techniques utilize past population's representative as initial individuals for MOEA.

Due to space constraints we move to part II [14] the mathematical formulation and parameter setting of APP; the utilization of serial sequence generation scheme (SSGS) as the scheme for our schedule generation [15]; and the innovation of cross-over and mutation operators to become suitable for NSGA-II. This part I is then organized as follows: CBA is introduced in Section II, and McBA in section III. A case study is presented in section IV where various types of experiments showed the superiority of McBA over other techniques. The last section is devoted to the conclusion of this part.

II. Centroid-Based Adaptation

As in many other EA techniques, MOEA's convergence towards the optimal solution is significantly influenced by its initial population. To exploit the existence of MOEA's non-dominated set after each change our previous paper [1] introduced the concept of centroid which embodies the set. Instead of using the entire set from previous change its centroid will be used as part of the initial population for MOEA's search of new optimal solution to adapt to current change.

Basing from biology a chromosome is composed of genes formed as string. Evolutionary Algorithm emulates this by representing chromosomes and genes by its solutions and objects, respectively [4]. Chromosome will be referred as individual and set of chromosomes as population, from here onwards. Let $i = 1, ..., n$ be the gene index in a chromosome with n number of genes; $P(t)$ be the set of solutions $x^j(t)$ for the environmental change indexed t; j the solution index; and $P_{nd}(t)$ be the set of non-dominated solutions in $P(t)$, i.e. $P_{nd}(t) \subseteq P(t)$. We define in [1] the centroid's gene at i as,

$$C_i(t) = \frac{1}{N_d} \sum_{x^j(t) \in P_{nd}(t)} x_i^j(t) \tag{1}$$

where $x_i^j(t)$ as the gene's attribute at (i, j), $N_d = |P_{nd}(t)|$, and $C(t)$ as the centroid for $P_{nd}(t)$.

In the implementation of our methodology, genes' attribute is the task ID number and the chromosome represents a schedule of tasks. Given a population $P_{nd}(t)$ of chromosomes a sample distribution of these IDs is shown in Figure 1 where each vertical strip signify an ID distribution. For a given gene index i denoted by the horizontal axis and ID d denoted by the vertical axis, each block in this strip is the number of genes – expressed by the grey level – with ID equal to d. Dividing this number by $|P_{nd}(t)|$ yields a function that is assumed to be a Probability Density Function (PDF) $P(i, d)$ of the distribution. The symbol "*" in the figure denotes $C_i(t)$ which, for a fixed t will be shortened to C_i. The figure illustrates that C_is are not far from the most popular ID at gene i.

Note that a centroid, which represents a schedule, is not necessarily task-precedence-feasible. Centroid's genes in infeasible centroid are repaired with minimal perturbation as follows: Consider a feasible section of the schedule/chromosome comprised with centroid's genes having indices $1 \le k < i$ where i is the current gene centroid under consideration. Now if the i^{th} gene centroid will make the section $1 \le k \le i$ feasible then its ID will not be replaced. Otherwise, its ID will be replaced by another which can make the latter section feasible and which is nearest to it. For example, suppose the first two centroid's genes are made feasible already and that C_3 with ID of 3 does not make the first to third centroid's genes feasible. Further, suppose there are unassigned IDs 1, 4, and 7 which are feasibly suitable replacements for C_3. Then ID of 4 (closest to ID number 3) will replace that of C_3. Let the function $\mathcal{F}(C_i)$, called Minimal Repairer, denotes this action of transforming infeasibility-causing gene to a feasibly suitable one.

The current environmental change can render all previous centroids $C(t)$ to be infeasible. Some of $C(t)$'s genes $C_i(t)$ will be repaired to become $C_i'(t)$ as described in Equation 2,

$$C_i'(t) = \begin{cases} C_i(t) & C_i(t) \text{ is precedence feasible} \\ \mathcal{F}(C_i(t)) & \text{otherwise} \end{cases} \tag{2}$$

The initial population $P_0(M)$ utilized by MOEA to find the solution for the current environmental change, indexed M, is defined as,

$$P_0(M) = C^{new} \cup P'(M-1) \tag{3}$$

where $C^{new} = \cup_{t=t_o}^{M-1} C'(t)$, $t_o = \max\{M - N_c, 1\}$, N_c is the maximum number of centroids, $P'(M-1)$ is the set of $N - |C^{new}|$ solutions randomly chosen from $P(M-1)$.

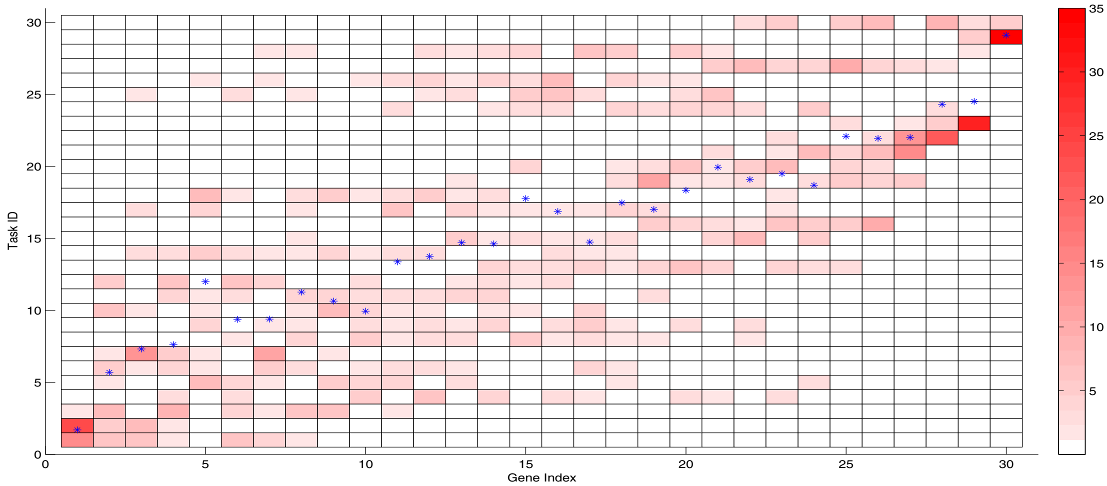

Fig. 1. Centroids prior to insertion of new tasks

III. Mapping Task ID for Centroid-Based Adaptation (MCBA)

One of the task precedence graph employed in this paper is shown in Figure 4 where the numbers inside the blocks are the task ID numbers, "S" the schedule start, and "E" the schedule end. Prior to addition of new tasks, only the ID's inside the rectangles are present which are chosen to be proportional to the order of precedence. This arrangement will result to centroid's genes being near the most number of task IDs as depicted in Figure 1. These ID numbers must be preserved when additional tasks are added to avoid confusing administration officers who will analyse schedules/solutions. For the case study in Section IV, there will be 10 new tasks inserted; enclosed by a circle in Figure 4 with IDs 31 to 40. The side-effect to maintaining old task IDs after the insertion is the shift of centroid's genes far from the most popular task ID of its associated distribution. This shift is evident in Figure 2 where at gene index 9 the most popular ID is 33 while the centroid is at 19.

To formally prove this shift: Let $P_1(i,d)$ be the $P(i,d)$ before new tasks are inserted; $\mathbf{E}$ be the expectation operator; and $\mathbf{E}[dP_1(i,d)] = \overline{d_1}$. Suppose with the insertion a new peak appears due to a new PDF $P_2(i,d)$, with $\mathbf{E}[dP_2(i,d)] = \overline{d_2}$, superimposed on $P_1(i,d)$. The overall PDF changes to,

$$P(i,d) = \alpha P_1(i,d) + \beta P_2(i,d) \tag{4}$$

where $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$. Considering that the expectation of any PDF is unity then,

$$\mathbf{E}[P(i,d)] = \alpha\mathbf{E}[P_1(i,d)] + \beta\mathbf{E}[P_2(i,d)] = 1 = \alpha + \beta \tag{5}$$

Now, taking the average with respect to ID,

$$\mathbf{E}[dP(i,d)] = \alpha\mathbf{E}[dP_1(i,d)] + \beta\mathbf{E}[dP_2(i,d)] \tag{6}$$

$$\overline{d} = \alpha\overline{d_1} + \beta\overline{d_2} \tag{7}$$

The absolute shift of centroid is, using Equation 5,

$$\Delta C = \left|\overline{d} - \overline{d_1}\right| = \left|\overline{d_2} - \overline{d_1}\right|\beta \tag{8}$$

which implies that the shift is largest when the two ID averages $\overline{d_1}$ and $\overline{d_2}$ are farthest.

Centroids are suppose to faithfully represent the distribution of IDs in solutions. However, as just discussed the addition of new tasks makes this untenable. The approach taken in this paper is to map IDs, during computation, such that resulting IDs are proportional to its order of precedence. For instance in Figure 4, the IDs in the first order of precedence are 1, 2, and 3 which were mapped to same value; while in the second order of precedence are 5, 10, 14, 6, 7, 12, 31, 32, and 33 which were mapped to 5, 10, 14, 6, 7, 12, 8, 9, and 11 respectively. Let the function $\mathcal{C}(d)$, where d is the original ID, perform this mapping operation. This function enables the system to faithfully represent the distribution of IDs in the solution as depicted in Figure 3 with 10 new genes inserted to the original 30. The mapped IDs will be restored to its original values after finding the optimal solution via MOEA.

Another side-effect of adding new tasks to the system is to render previous population $P(t)$ in Equation 1 useless, since its individuals/chromosomes has lesser number of genes than what the current state of the system required. To remedy, new genes corresponding to new tasks are inserted to their immediate predecessors for all chromosomes in $P(t), 1 \leq t \leq M-1$, then centroids of $P(t)$ are recomputed via the mapping $\mathcal{C}$. Gene insertion is only performed when new tasks are added to the system.

The approach of Minimal Repairer $\mathcal{F}$ defined in Section II; the mapping operation $\mathcal{C}$; and the insertion of new genes to all chromosomes is called, Mapping Task IDs for Centroid-Based Adaptation (McBA).

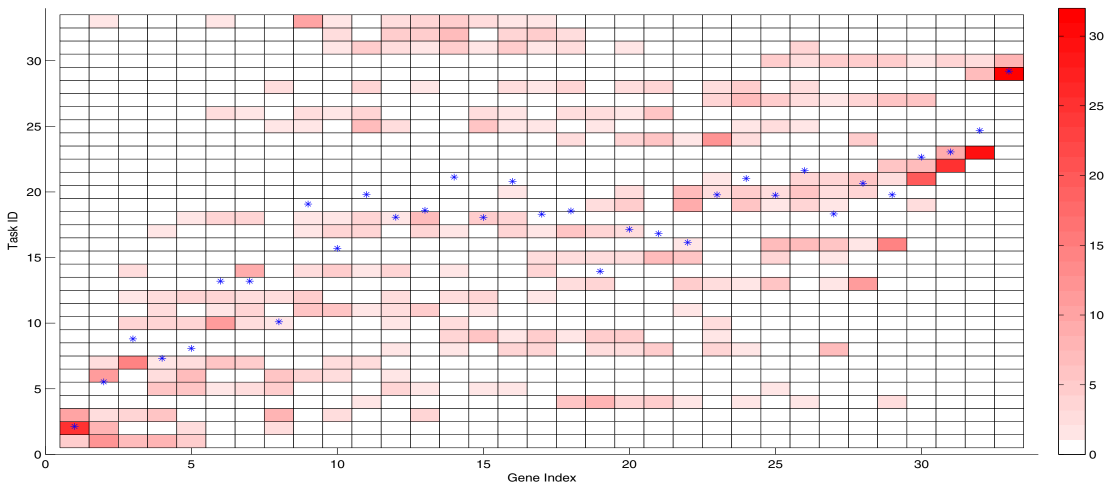

Fig. 2. Centroid far from most popular ID

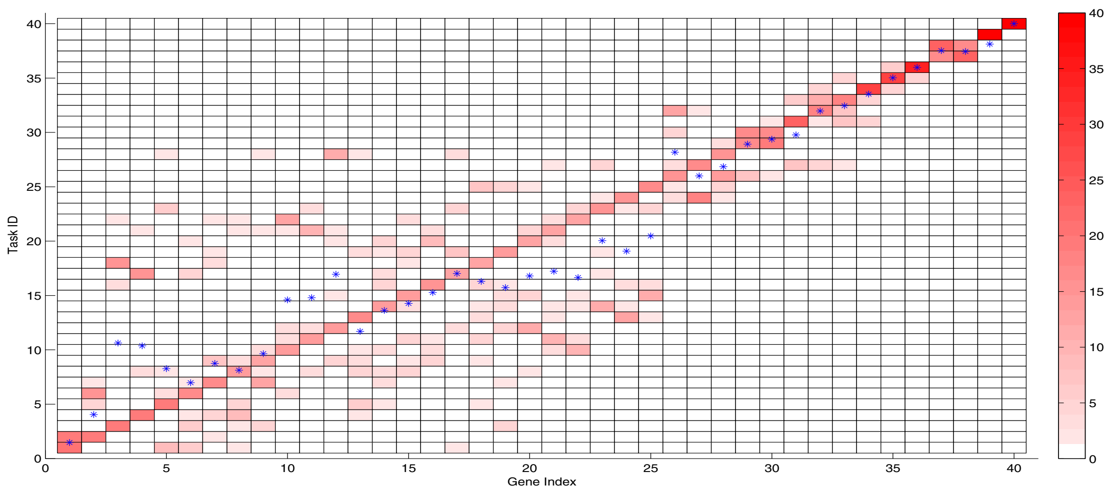

Fig. 3. Centroid for Mapped IDs

IV. A Case Study

Detailed description on the case study for this paper is in the second part. This case study is a general scheduling problem adapted from our previous publication [1]. It has 30 starting tasks with precedence relationship given in Figure 4 when circular figures are replaced by arcs. However, being obliged to adapt to new situation during the execution of the plan, 10 more tasks were added, enclosed by circle.

A. Validating Methods

In Section III, we described the McBA technique. Its performance is validated through four other methods [1] adjusting from adaption. All of them does not implement task ID mapping as in $\mathcal{C}$, except McBAR described below, but implement gene insertion for new tasks when there is a change in task number, except for Random Initialization technique, also described below.

- McBA with randomization (*McBAR*) employs McBA with $P'(M-1)$ in Equation 3 randomly generated
- Centroid-Based Adaptation (*CBA*) employs centroid defined in Equation 1; and randomly selects feasible gene instead of using $\mathcal{F}$.
- Last Population (*LPOP*) uses the last population obtained from the previous adaption period.
- Non-Dominated solutions from the Last Population (*NDLPOP*) uses the non-dominated previous solutions with dominated solutions being replaced by randomly initialized chromosome.
- Random Initialization (*RI*) creates new population by random initialization without caring any information in

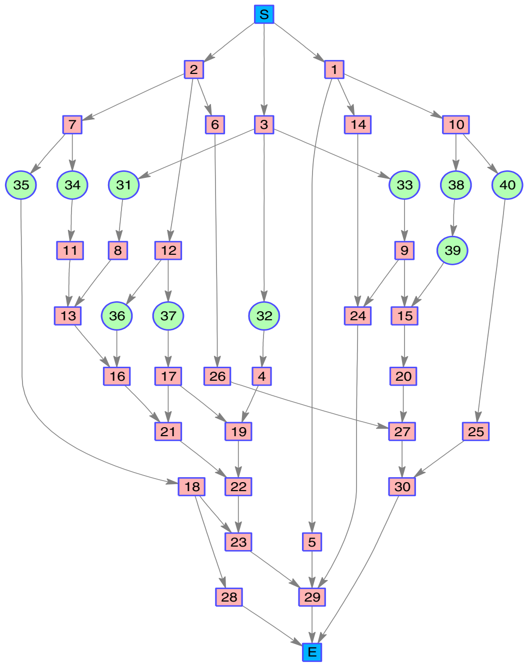

Fig. 4. Fourth task addition

the past.

Each of these methods used different starting population to compute their respective baselines. Further, the initial population employed by each method to compute their respective optimal population for a current change is related to their respective optimal population for the previous changes – except in *RI* – and not to that of other methods. In the contrary, in the second part of this paper, the best optimal population of the previous change among those produced by all methods will be employed as initial population for all methods to find their respective optimal solutions for the current change.

B. Experiments

McBA is tested for Nine experiments where there are Twelve environmental changes occurring at times in the set $T_d = \{4, 6, 8, 12, 13, 16, 19, 23, 26, 30, 33, 37\}$. Task number changes occurred at times in the set $N_t = \{8, 19, 26, 33\}$, i.e. at 3^{rd}, 7^{th}, 9^{th}, and 11^{th} order of changes where there are 3, 2, 2, and 3, added new tasks respectively. Resource availability changes at times in the set $R_s = \{6, 13, 23, 37\}$, i.e. at 2^{cd}, 5^{th}, 8^{th}, and 12^{th} order of changes. These sets are related in Table I where $\times$ and $\checkmark$ denote that the type of change is not applied and applied, respectively. Variation in original task duration is by the addition of normally distributed random value with a standard deviation of δ. The types of tests/experiments are as follows:

1) Resource availability and task number change only
2) Task Duration, with $\delta = 3.0$, changing at times in $T_d - N_t$ and task number changing at times in N_t, i.e., not simultaneous with task duration change

TABLE I
PERIOD OF ENVIRONMENTAL CHANGE

Order of Change	T_d	N_t	R_s
1	4	$\times$	$\times$
2	6	$\times$	$\checkmark$
3	8	$\checkmark$	$\times$
4	12	$\times$	$\times$
5	13	$\times$	$\checkmark$
6	16	$\times$	$\times$
7	19	$\checkmark$	$\times$
8	23	$\times$	$\checkmark$
9	26	$\checkmark$	$\times$
10	30	$\times$	$\times$
11	33	$\checkmark$	$\times$
12	37	$\times$	$\checkmark$

3) Task Duration, with $\delta = 6.0$, changing at times in $T_d - N_t$ and task number changing at times in N_t
4) Task Duration, with $\delta = 3.0$, changing at times in T_d and task number changing simultaneously at times in N_t
5) Task Duration, with $\delta = 6.0$, changing at times in T_d and task number changing simultaneously at times in N_t
6) Task Duration, with $\delta = 3.0$, changing at times in $T_d - N_t - R_s$, resource changing at times in R_s, and task number changing at times in T_d, i.e., all types of changes are non-simultaneous
7) Task Duration, with $\delta = 6.0$, changing at times in $T_d - N_t - R_s$, resource changing at times in R_s, and task number changing at times in T_d
8) Task Duration, with $\delta = 3.0$, changing at times in $T_d - R_s$, task number changing at times in N_t, i.e., simultaneous with duration change; resources changing at times in R_s
9) Task Duration, with $\delta = 6.0$, changing at times in $T_d - R_s$, task number changing at times in N_t; resources changing at times in R_s

A sample optimal schedule in the eleventh change is shown in Figure 5 where the horizontal axis is time; the numbers in the rectangular strips are the task numbers; rectangular length is the task duration; and Rs are the resources. Notice the presence of new tasks with IDs 31 to 40 which illustrate the effect of inserting new tasks to the system and demonstrate how schedule was adapted to cope with changes; thereby, demonstrating the adaptability of the system.

C. Comparison of McBA to Other Methods

We use the measure called "set coverage" (SC) [16] to access the performance of McBA compared to the validating methods. SC is determined between two sets A and B (SC(A,B)) by counting the number of solutions in B that are dominated by solutions in A:

$$SC(A, B) = \frac{|b \in B| \exists a \in A : a \preceq b|}{|B|} \quad (9)$$

where $a \preceq b$ indicates a dominates b.

Each experiment is run, of similar experiment type, repeatedly for 30 times with each run having different additional random value to the original task duration in every change

Fig. 5. Chosen Individual at Change 12

in task duration. The set coverage was computed for all the 30 runs and its average is tabulated in Table II, however, only for selected methods and changes. In each entry the first value is the mean set coverage between the method in the left column against and the method in the uppermost row. The next value in the entry, separated by $\pm$ symbol from the first, is the standard error from all the runs. The first and second columns in this table correspond to experiment and change numbers, respectively.

Let technique X be called superior to Y if $SC(X, Y) > SC(Y, X)$, and inferior otherwise. It is clear from all results – not shown – that McBA and its variant McBAR are superior to other methods on all experiments and in all changes; except at 11^{th} change of experiments 8 and 9 and 7^{th} change of experiment 9 for McBA; and 7^{th} change of experiment 9 for McBAR. Data for these exceptions are highlighted in the table. Further, exception is true at the first and second changes for all experiments where there is only identity mapping ($\mathcal{C}(i) = i$), hence made McBA's no different to CBA. More so, there are only one and two centroids at first and second changes respectively, rendering McBA and McBAR ineffective.

In the foregoing discussion, the conclusions cited are valid only for the experiments described in this paper:

The task duration change amplitude δ is equal to 3 in Experiments 2 and 6 while it is 6.0 in Experiments 3 and 7. Task duration and number do not change simultaneously in these experiments. Performances of McBAs (McBA and McBAR) in all these experiments are superior to all other techniques. However, McBA's performance (signified by the highlighted values in Table II) becomes inferior to RI in Experiment 8 (where $\delta = 3.0$) and in Experiment 9 (where $\delta = 6.0$) at the 7^{th} and 11^{th} changes and McBAR becomes inferior in Experiment 9 at the 7^{th} change. There are simultaneous changes in task duration and number at these orders of change. This showed that McBA's performance is not much influence by the amount of δ but rather by the simultaneity. And that McBAR is affected strongly with the amount of δ and the simultaneity. The underlying cause of this inferiority is the topic of part II.

The number of tasks at 7^{th} and 11^{th} changes of Experiments 8 and 9 are increased by 2 and 3, respectively. Similar situation happens in Experiments 2 to 7 which, however, showed good performances of McBAs. Thus, it could be said that the change in task number has a strong influence on the performances of both McBA and McBAR only when there is a simultaneous change in task duration and number.

In Experiments 8 and 9, there is an increase in task number by 2 at the 7^{th} and 9^{th} changes and by 3 at 3^{rd} and 11^{th} changes. While McBAs' performances are inferior to RI at 7^{th} and 9^{th} changes they remain superior to all other techniques at the 3^{rd} and 9^{th} changes despite correspondingly similar amount of change in task numbers. This could be due to the change of resource availability at the 2^{cd} and 8^{th} changes which, in this case supports rather than degrades McBAs' performances at the 3^{rd} and 7^{th} changes. This showed the influence of previous events in the chain of changes to McBAs' performances. Due to this influence, part II will revise selection of initial population described in Equation 3.

Experiments 4 and 5 has simultaneous change in task duration and number but does not have change in resource availability. In these experiments McBAs are superior to other techniques. However, not in Experiments 8 and 9 which are correspondingly similar to Experiments 4 and 5 except for having changes in resource availability. Experiment 1 does have change in resource availability and task duration and yet McBAs perform well. These results suggest that McBAs performances are strongly influenced by resource availability change only when there is a simultaneous change in task duration and number.

As a summary, experiments showed that performances of

TABLE II
SELECTED SET COVERAGES

#	Order	Technique	CBA	MBA	MBAR	LPOP	NDLPOP	RI
1	7	MBA	0.5483 ± 0.0840	N/A	0.2942 ± 0.0748	0.5200 ± 0.0846	0.4767 ± 0.0857	0.5217 ± 0.0838
		MBAR	0.7250 ± 0.0777	0.4617 ± 0.0812	N/A	0.7558 ± 0.0782	0.7042 ± 0.0759	0.6858 ± 0.0778
		RI	0.4675 ± 0.0854	0.2117 ± 0.0636	0.0517 ± 0.0269	0.4642 ± 0.0835	0.5225 ± 0.0816	N/A
	11	MBA	0.5917 ± 0.0702	N/A	0.1667 ± 0.0545	0.6342 ± 0.0726	0.6508 ± 0.0730	0.4642 ± 0.0675
		MBAR	0.8325 ± 0.0517	0.6967 ± 0.0654	N/A	0.8700 ± 0.0461	0.9042 ± 0.0408	0.7775 ± 0.0566
		RI	0.5142 ± 0.0749	0.1917 ± 0.0475	0.0433 ± 0.0239	0.5517 ± 0.0740	0.5725 ± 0.0764	N/A
2	7	MBA	0.4192 ± 0.0841	N/A	0.2875 ± 0.0767	0.5425 ± 0.0880	0.5408 ± 0.0890	0.4217 ± 0.0817
		MBAR	0.4400 ± 0.0892	0.2817 ± 0.0792	N/A	0.5733 ± 0.0858	0.5733 ± 0.0873	0.5417 ± 0.0824
		RI	0.3392 ± 0.0797	0.0975 ± 0.0438	0.1308 ± 0.0507	0.3267 ± 0.0822	0.4408 ± 0.0825	N/A
	11	MBA	0.6967 ± 0.0634	N/A	0.3775 ± 0.0772	0.6917 ± 0.0685	0.7542 ± 0.0678	0.5925 ± 0.0795
		MBAR	0.9133 ± 0.0400	0.4150 ± 0.0757	N/A	0.7842 ± 0.0630	0.8533 ± 0.0507	0.6775 ± 0.0683
		RI	0.6208 ± 0.0713	0.3033 ± 0.0717	0.1450 ± 0.0471	0.6850 ± 0.0633	0.7467 ± 0.0542	N/A
3	7	MBA	0.3267 ± 0.0806	N/A	0.2783 ± 0.0716	0.4125 ± 0.0830	0.4200 ± 0.0832	0.3467 ± 0.0823
		MBAR	0.6033 ± 0.0849	0.3483 ± 0.0831	N/A	0.6042 ± 0.0805	0.7258 ± 0.0730	0.5967 ± 0.0781
		RI	0.3658 ± 0.0834	0.2117 ± 0.0701	0.1167 ± 0.0538	0.4250 ± 0.0843	0.2900 ± 0.0806	N/A
	11	MBA	0.5792 ± 0.0778	N/A	0.2792 ± 0.0671	0.5892 ± 0.0759	0.7133 ± 0.0688	0.5117 ± 0.0775
		MBAR	0.7000 ± 0.0708	0.4142 ± 0.0770	N/A	0.7450 ± 0.0662	0.8050 ± 0.0655	0.6425 ± 0.0708
		RI	0.5058 ± 0.0755	0.3275 ± 0.0764	0.1550 ± 0.0556	0.5242 ± 0.0804	0.6433 ± 0.0675	N/A
4	7	MBA	0.4658 ± 0.0807	N/A	0.1333 ± 0.0446	0.4917 ± 0.0827	0.4858 ± 0.0830	0.4300 ± 0.0839
		MBAR	0.7333 ± 0.0757	0.4983 ± 0.0772	N/A	0.7917 ± 0.0667	0.8108 ± 0.0691	0.6267 ± 0.0815
		RI	0.3475 ± 0.0816	0.2125 ± 0.0591	0.1033 ± 0.0468	0.3808 ± 0.0811	0.4850 ± 0.0831	N/A
	11	MBA	0.5042 ± 0.0845	N/A	0.2533 ± 0.0648	0.7117 ± 0.0802	0.6133 ± 0.0877	0.5725 ± 0.0897
		MBAR	0.6842 ± 0.0784	0.4033 ± 0.0796	N/A	0.8317 ± 0.0691	0.8192 ± 0.0695	0.7250 ± 0.0757
		RI	0.2450 ± 0.0721	0.1267 ± 0.0422	0.1133 ± 0.0473	0.4033 ± 0.0834	0.4350 ± 0.0796	N/A
5	7	MBA	0.6275 ± 0.0827	N/A	0.3733 ± 0.0790	0.6433 ± 0.0839	0.7100 ± 0.0772	0.5850 ± 0.0831
		MBAR	0.6450 ± 0.0835	0.4083 ± 0.0763	N/A	0.6342 ± 0.0842	0.6975 ± 0.0788	0.6575 ± 0.0764
		RI	0.4150 ± 0.0787	0.2008 ± 0.0642	0.1617 ± 0.0579	0.3567 ± 0.0763	0.4608 ± 0.0849	N/A
	11	MBA	0.4708 ± 0.0797	N/A	0.2892 ± 0.0729	0.5442 ± 0.0787	0.5167 ± 0.0778	0.3775 ± 0.0757
		MBAR	0.6808 ± 0.0800	0.5375 ± 0.0778	N/A	0.7525 ± 0.0677	0.7233 ± 0.0721	0.5675 ± 0.0788
		RI	0.5217 ± 0.0736	0.3700 ± 0.0731	0.2200 ± 0.0608	0.5917 ± 0.0692	0.6083 ± 0.0720	N/A
6	7	MBA	0.5056 ± 0.0849	N/A	0.1887 ± 0.0552	0.3952 ± 0.0818	0.6968 ± 0.0787	0.4637 ± 0.0730
		MBAR	0.5879 ± 0.0792	0.4258 ± 0.0830	N/A	0.6419 ± 0.0843	0.6710 ± 0.0750	0.6887 ± 0.0699
		RI	0.3081 ± 0.0730	0.1008 ± 0.0425	0.0895 ± 0.0435	0.3597 ± 0.0779	0.5613 ± 0.0831	N/A
	11	MBA	0.5129 ± 0.0847	N/A	0.1129 ± 0.0457	0.4694 ± 0.0842	0.5258 ± 0.0813	0.4887 ± 0.0850
		MBAR	0.7573 ± 0.0694	0.5218 ± 0.0807	N/A	0.7887 ± 0.0674	0.6331 ± 0.0855	0.8645 ± 0.0518
		RI	0.2935 ± 0.0771	0.0806 ± 0.0337	0.0266 ± 0.0271	0.4113 ± 0.0845	0.4363 ± 0.0868	N/A
7	7	MBA	0.4208 ± 0.0853	N/A	0.1417 ± 0.0511	0.4183 ± 0.0826	0.4350 ± 0.0838	0.4075 ± 0.0789
		MBAR	0.5567 ± 0.0810	0.4892 ± 0.0791	N/A	0.5642 ± 0.0801	0.6158 ± 0.0795	0.5792 ± 0.0814
		RI	0.4283 ± 0.0838	0.3342 ± 0.0756	0.2258 ± 0.0612	0.4167 ± 0.0767	0.3150 ± 0.0779	N/A
	11	MBA	0.3317 ± 0.0840	N/A	0.0758 ± 0.0355	0.3817 ± 0.0820	0.4000 ± 0.0826	0.4033 ± 0.0764
		MBAR	0.6450 ± 0.0818	0.5467 ± 0.0778	N/A	0.6067 ± 0.0803	0.6633 ± 0.0722	0.7475 ± 0.0712
		RI	0.2050 ± 0.0671	0.2092 ± 0.0659	0.0733 ± 0.0366	0.2250 ± 0.0638	0.2650 ± 0.0699	N/A
8	7	MBA	0.1950 ± 0.0652	N/A	0.2050 ± 0.0634	0.3483 ± 0.0841	0.4333 ± 0.0882	**0.3342** ± 0.0778
		MBAR	0.3917 ± 0.0755	0.4417 ± 0.0877	N/A	0.5175 ± 0.0868	0.5383 ± 0.0870	**0.3700** ± 0.0765
		RI	0.3867 ± 0.0727	**0.3342** ± 0.0805	**0.1642** ± 0.0638	0.2900 ± 0.0769	0.4700 ± 0.0812	N/A
	11	MBA	0.5325 ± 0.0748	N/A	0.2042 ± 0.0608	0.4450 ± 0.0772	0.4858 ± 0.0827	**0.3400** ± 0.0645
		MBAR	0.5942 ± 0.0763	0.5867 ± 0.0751	N/A	0.6208 ± 0.0690	0.6433 ± 0.0747	**0.4375** ± 0.0736
		RI	0.5300 ± 0.0813	**0.4592** ± 0.0800	**0.3425** ± 0.0760	0.5508 ± 0.0783	0.6375 ± 0.0720	N/A
9	7	MBA	0.2525 ± 0.0757	N/A	0.1783 ± 0.0591	0.3258 ± 0.0786	0.4633 ± 0.0806	**0.1167** ± 0.0521
		MBAR	0.2475 ± 0.0773	0.2542 ± 0.0685	N/A	0.4608 ± 0.0889	0.4700 ± 0.0870	**0.2550** ± 0.0695
		RI	0.3700 ± 0.0844	**0.3108** ± 0.0806	**0.3700** ± 0.0761	0.5000 ± 0.0888	0.5492 ± 0.0825	N/A
	11	MBA	0.4500 ± 0.0809	N/A	0.3883 ± 0.0742	0.5567 ± 0.0776	0.7125 ± 0.0716	**0.3292** ± 0.0754
		MBAR	0.4425 ± 0.0765	0.3825 ± 0.0750	N/A	0.6792 ± 0.0730	0.6442 ± 0.0802	**0.4025** ± 0.0733
		RI	0.5725 ± 0.0715	**0.3700** ± 0.0746	**0.3225** ± 0.0712	0.7025 ± 0.0630	0.7100 ± 0.0752	N/A

McBAs are strongly influenced by the amplitude of change in task duration and amount of change in task numbers given that these two parameters change simultaneously and that changes in resource availability occur somewhere else in the sequence of changes.

McBAR employs the centroids and randomly generated individuals as initial population for MOEA while McBA employs the centroids and all of elements of $P'(M-1)$ in Equation 3. Results showed that in general, McBAR outperforms McBA showing the irrelevance of the previous population $P'(M-1)$.

V. Conclusion

In this paper, we introduced the concept of mapping task-ID prior to the computation of centroid of the non-dominated set and the insertion of new tasks to optimal individuals from previous environmental changes. This technique proves to be extremely helpful in finding optimal schedules for the validating mission planning case study, especially on the occurrence of new tasks. Further, experiments showed that performance of this technique is strongly influenced by the amplitude of change in task duration and amount of change in task numbers given that these two parameters change simultaneously and that changes in resource availability occur somewhere else in the sequence of changes.

ACKNOWLEDGEMENTS

This work is supported by Australian Research Council (ARC) Grant DP0985723 and by grants N 516 384734 and N 519 578038 from the Polish Ministry of Science and Higher Education (MNiSW). The second author acknowledged the funding from Vietnam's National Foundation for Science and Technology Development (NAFOSTED), Grant No 102.01-2010.12.

REFERENCES

[1] L. T. Bui, Z. Michalewicz, E. Parkinson, and M. B. Abello, "Adaptation in Dynamic Environments: A Case Study in Mission Planning," *IEEE Transactions on Evolutionary Computation*, 2011.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[3] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.

[4] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - a survey," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[5] J. Branke, *Evolutionary optimization in dynamic environments*. Massachusetts USA: Kluwer Academic Publishers, 2002.

[6] S. Van de Vonder, E. Demeulemeester, and W. Herroelen, "A classification of predictive-reactive project scheduling procedures," *Journal of Scheduling*, vol. 10, no. 3, pp. 195–207, 2007.

[7] L. T. Bui, M. Barlow, and H. A. Abbass, "A Multiobjective Risk-based Framework for Mission Capability Planning," *New Mathematics and Natural Computation, World Scientific*, vol. 5, no. 2, pp. 459–485, 2009.

[8] C. Ramsey and J. Grefenstette, "Case-based initialization of genetic algorithms," in *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 84–91.

[9] D. Goldberg and R. Smith, "Nonstationary function optimization using genetic algorithms with dominance and diploidy," in *Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 59–68.

[10] G. Barlow and S. Smith, "A memory enhanced evolutionary algorithm for dynamic scheduling problems," in *Applications of Evolutionary Computing: EvoWorkshops 2008*, M. G. et al., Ed., vol. 4974/2008. Berlin / Heidelberg: Springer, April 2008, pp. 606–615.

[11] A. J. Page and T. J. Naughton, "Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing," in *Proceedings of the 19th IEEE/ACM International Parallel and Distributed Processing Symposium*. Denver, Colorado, USA: IEEE Computer Society, April 2005.

[12] C. Artigues and F. Roubellat, "A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes," *European Journal of Operational Research*, vol. 127, no. 2, pp. 297–316, 2000.

[13] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *J. Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.

[14] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An Adaptive Approach for Solving Dynamic Scheduling with Time-varying Number of Tasks – Part II," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, 2011.

[15] E. Demeulemeester and W. Herroelen, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem," *Management Science*, vol. 38, no. 12, pp. 1803 – 1818, 1992.

[16] K. Deb, "Multiobjective genetic algorithms: Problems difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.