WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

A Reactive-Proactive Approach for Solving Dynamic Scheduling with Time-varying Number of Tasks

Manuel Blanco ABELLO*, Zbigniew MICHALEWICZ†, Lam Thu BUI‡
*School of Computer Science, University of Adelaide, Adelaide, South Australia
Email: manuel.abello@adelaide.edu.au
‡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
†Le Quy Don Technical University, Viet Nam
Email: lam.bui07@gmail.com

***Abstract*—Any system (whether in the area of finance, manufacturing, administration, etc.) that operates in a dynamic environment needs to be adaptive to changes; it should also anticipate possible adverse events to remain competitive. In our previous research in this area we experimented with one particular approach: Mapping of Task ID for Centroid-Based Adaptation with Random Immigrants (McBAR) to address problems of environmental changes for Resource-Constrained Project Scheduling (RCPS) problem, especially when the latter involves changes in task numbers. However, at that time, McBAR was applied as reactive tool only. In this paper we extend McBAR approach to the RCPS problem in a proactive-reactive way. The system handles also three competing objectives: cost, makespan, and the risk of failure.**

We have not found any papers that deal with risk on the RCPS problem and utilize the attributes of plans from the past environmental changes. This particular aspect is incorporated in McBAR – experimental results indicate the efficiency of such approach in finding optimal solutions for a current change. In this paper we also analyze, under the effects of environmental dynamics, the variation of risk computed via McBAR and of parameters related to optimization. Further, we compare McBAR to other Evolutionary Algorithm approach in the same problem.

***Index Terms*—adaptation, dynamic environments, multi-objective optimization, risk management**

I. INTRODUCTION

In most decision-support systems that deal with planning problems, risk must be addressed explicitly. Better plans (with respect to some key performance indicators) might be rejected due to high risk associated with them. The reason is uncertainty of the input data and variability of processes - elements which are present in most businesses and industries. Thus it is crucial to consider risk in planning activities.

Clearly, the values of environmental variables often cannot be anticipated during creation of a plan and when an unexpected event occurs, the current plan may no longer be optimal or even feasible [1]. Thus, it is necessary to revise this plan to cope with the new state of the environment. Many real-world problems are set in this type (dynamic, non-stationary) of environment, where objectives, constraints, or even variables may change in time. In this paper we propose a combination of proactive and reactive approaches to address the issue of finding and modifying optimal plans.

There have been various approaches in planning that considered risk. The work of Bui, et. al [2] was proactive in its approach, but did not consider dynamic environment. However, the authors dealt with the Resource Constrained Project Scheduling (RCPS) problem with three conflicting objectives: cost, makespan and risk. In finance, several authors employed stochastic programming [3], [4]. Hochreiter [5] and Tometzki and Engel [6] dealt with multi-objective problems involving risk. Several authors dealt with the problem of dynamic environment involving risk but not on the RCPS problem; e.g. [7], [8] to name a few. In particular, Ouarda and Labadie [9] employed Lagrangian function for a dynamic multi-reservoir system. Lambrechts, et. al, [1] enumerated strategies on proactive and reactive approaches to the RCPS problems. RCPS with multiple-objectives but without considering risk nor dynamic environments are found in [10], [11] and in [12] which dealt with three objectives – makespan, cost, and tardiness. However, direct application of memory-based Evolutionary Algorithm (EA), to the best of our knowledge, is absent in these areas.

This paper applies our previous approach, Mapping of Task-ID for Centroid-Based Adaptation with Random Immigrants (McBAR) in a proactive and reactive manner to the RCPS problem. In this application, a risk-anticipative baseline schedule is first made and then revised when changes occur, with each revision still anticipating risk for the remaining unimplemented components while preserving on-going or finished components of the schedule from previous change. In this way, the search for new optimal schedule is still geared towards lower risk despite environmental changes. McBAR is a memory-based EA which was proven in our previous works [13], [14] to be versatile in dynamic environment, especially when task number varies in time.

The main contribution of this paper is the analysis, under the effects of environmental dynamics in the RCPS problem, on the variation of risk computed via McBAR with respect to changes in task duration, task number, and resource availability. We also analyze variation of parameters related to optimization such as solution convergence and hypervolume. Further, we compare the performance of McBAR to another technique in EA.

The paper is organised as follows. A brief review on schedule adaptation in dynamic environment is discussed in Section II, followed by discussion on risk in Section III. This is followed by a brief description of multi-objective problems in Section IV and then by the description of the McBAR approach in Sections V and VI. Next, the mathematical model for the convergence of hypervolume with respect to number of generations in Evolutionary Algorithm is presented in Section VII. The experimental set-up is described in Section VIII and the experimantal results are discussed in Section IX. Finally, the conclusion and future works are discussed in Section X.

II. Adaptation in Dynamic Environment

As stated in the Introduction, execution of plans can encounter various disruption and there might be a need to revise these plans. This scenario belongs to a class of so-called Dynamic Optimization Problems (DOPs) [15], [16]. Approaches to DOPs could be categorize as follows:

1) **Reactive**: Methods in this category employ a solution found through some optimization and is used as baseline. This baseline is then revised to become optimally suitable to the new state of the environment whenever the latter changes. The revision does not consider future uncertainties. Further, revision of the solution components which are in-progress might be extremely expensive and hence must be preserved.
2) **Proactive**: In this category, methods anticipate future uncertainties. This anticipation could be based on some probabilistic distribution, or some other information, to search the optimal solution which, are robust under environmental changes [17], [18], [19]. The anticipation could be by adding slack time, resource slack, and cumulative instability weight [1]. Methods in this category may involve sensitivity analysis, such as Monte-Carlo simulation.
3) **No-baseline**: Unlike reactive methods, in this category baselines are not employed. Rather, every time environment changes, a new optimization search is perform to find a new adaptive solution.

Cumulative Instability Weight (CIW) described in [1] creates a schedule by arranging its components according to the decreasing order CIW. Instead of CIW parameter this paper employs probability of risk failure defined in Equation 1.

RCPS is a popular test environment in the study of dynamic environments [20]. Schedule in this context is composed of tasks obeying some precedence as illustrated in Figure 1. Further, active tasks are constrained not to exceed resource limit. To be compatible with RCPS, we modify MOEA's genetic operators as described in [13]. Problem formulation for this test environment is described in [21] added with probability of risk in defying resource constraint defined in Equation 1.

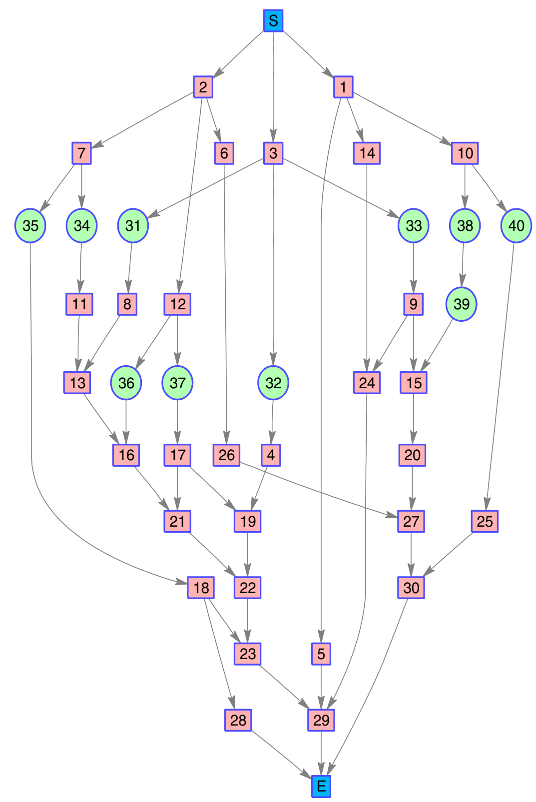

Fig. 1. Task Precedence: All new tasks present

III. Measure of Risk

In the literature, risk is measured in various ways: In a financial application, risk is measured as a monetary value lost after a given time bound [7]. In a fleet scheduling application, risk is the probability that a fleet would be unable to accommodate load or passengers [22]. An axiomatic approach was presented in [23] to define risk measure. In an RCPS application, risk was measured in [2] as the probability that a given plan will violate resource constraint when task durations are varied given that task starting times are fixed.

The last definition of risk measure is adapted in this paper which we now introduced. Let a plan be denoted as P_i. The probability Pf_i of a plan to violate resource constraint due to task duration variation is calculated by Monte Carlo simulation:

$$Pf_i = \frac{1}{T}\sum_{j=1}^{T} x_j^i \qquad (1)$$

where j is the simulation index whereby in this simulation tasks durations are varied randomly that is statistically independent across all tasks and all simulations; T is the number of simulations; and x_j^i represents whether the plan violates resource constraints or not and is determined as follows:

$$x_j^i = \begin{cases} 1 & \text{if } P_i \text{ violates the constraints} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

IV. Multi-Objective Optimization (MOO)

Another aspect to be taken into account in creating a plan is a possible presence of conflicting objectives, where it is necessary to weight one objective against the other: If a company desires to reduce production time it is likely to spend more. However, if it will reduce production cost, manufacturing time will be longer. Therefore, the two objectives cannot be reduced as low as desired at the same time.

Our previous works [13], [14] dealt with this issue by employing Multi-Objective Optimization using Evolutionary Algorithm (MOEA) which produces a set of solutions/plans/schedule, called Non-Dominated Set (NDS) [24], with comparable quality. A popular MOEA method is Non-dominated Sorting Genetic Algorithm II (NSGA2) [25]. No element of NDS excel in all objectives over other elements in the set. It is from this NDS that decision-makers may pick one to implement in the field. Elements of NDS lie in the hypersurface called Pareto Optimal Front (POF). For two solutions $\overrightarrow{x}_1$ and $\overrightarrow{x}_2$, we denote $\overrightarrow{x}_1 \preceq \overrightarrow{x}_2$ if $\overrightarrow{x}_1$ dominates $\overrightarrow{x}_2$.

It is essential to compare the performances of different methods. One way to make this comparison is through the Set Coverage. Given two sets A and B (each produced by the two methods to be compared), Set Coverage is found by counting the number of solutions in B that are dominated by solutions in A:

$$SC(A,B) = \frac{|b \in B| \ \exists \ a \ \in A : a \preceq b}{|B|} \quad (3)$$

Note that $SC(A,B)$ is not necessarily equal to $SC(B,A)$. In the foregoing, it will be assumed that $|A| = |B|$. If $SC(A,B) > SC(B,A)$ then there are more solutions in A that dominates those of B than the converse. For the former case, we say that the method producing A performs better or superior than that of B.

V. Centroid-Based Adaptation with Random Immigrants (CBAR)

Our previous works [13], [14], every time an environmental change occurs, MOEA is reapplied to find new set of NDS, S_{new}, optimally suitable for the new state of the environment. It is possible that S_{new} is strongly related to the set of NDS, S_{past}, corresponding to some past changes, such that it should be worthwhile to use the attribute of S_{past} as input information to MOEA to speed-up the search of S_{new}. In our previous work [13], the attribute employed was the centroid of NDS calculated for each S_{past}. It shows the overall tendency of its corresponding S_{past} towards S_{new}. This centroid, together with randomly generated plans, forms part of input information to MOEA to compute for S_{new}. This technique is called Centroid-Based Adaptation with Random Immigrants (CBAR). The said work showed CBAR's significant improvement over some techniques in finding better solutions under the effects of environmental dynamics.

Let a plan/schedule/solution be represented by a string of components which represent tasks in RCPS. In Evolutionary Algorithm terminology, this string represents chromosome and components represent genes; the chromosome is called individual and its set is called population. A preliminary centroid $C^p(t)$ will be defined prior to that of the actual centroid. This is a chromosome whose i^{th} gene is the mean [13],

$$C_i^p(t) = rnd\left(\frac{1}{N_d}\sum_{x^j(t)\in P_{nd}(t)} x_i^j(t)\right) \quad (4)$$

where $i = 1, \ldots, N_g$ is the gene index in a chromosome with N_g number of genes; $x^j(t)$ is an individual in a population $P(t)$ produced by MOEA associated to the t^{th} order of environmental change; j is the individual index; $x_i^j(t)$ is the gene's attribute at (i,j); $P_{nd}(t)$ is the set NDS in $P(t)$ (meaning, $P_{nd}(t) \subseteq P(t)$); $N_d = |P_{nd}(t)| > 1$; and $rnd()$ as the rounding operation of real-value to integer.

The preliminary centroid, in the context of RCPS, may not be necessarily task-precedence-feasible. Its infeasible genes are repaired as described in [13] employing a function $\mathcal{R}(C_i^p(t))$ that transforms an infeasible gene $C_i^p(t)$ to a feasible one. The preliminary centroid will be repaired to become $C(t)$ through,

$$C_i(t) = \begin{cases} C_i^p(t) & C_i^p(t) \text{ is precedence feasible} \\ \mathcal{R}(C_i^p(t)) & \text{otherwise} \end{cases} \quad (5)$$

where $C_i(t)$ is the i^{th} gene of the repaired centroid.

As cited in the previous section, MOEA requires an initial population $P_0(t)$ which our previous works employed, for the t^{th} environmental change,

$$P_0(t) \equiv C^{new} \cup R_{nd} \quad (6)$$

where $C^{new} = \cup_{k=t_o}^{t-1} C(k)$, $t_o = \max\{t - N_c, 1\}$, N_c is the maximum number of centroids, and R_{nd} is the set of $N - |C^{new}|$ solutions randomly generated, i.e., the random immigrants. This approach is called, Centroid-Based Adaptation with Random Immigrants (CBAR).

VI. Mapping Task ID for Centroid-Based Adaptation with Random Immigrants (McBAR)

Our previous work [13], dealt on changes in task duration, task precedence, and availability of resources in RCPS problems. In our preliminary investigation [14], we extended the types of changes to include the increase in the number of tasks, i.e., change in problem dimension. CBA is unsuitable for this type of change. We now discuss the remedy to this problem. Before proceeding, lets take the notation $P(t_o \leq k < t)$ to denote populations produced by MOEA from change order t_o to $t-1$ where t_o is defined in relation to Equation 6 constraining the number of centroids to at most N_c.

When for the first time new tasks (circular objects in Figure 1) are introduced to our RCPS system at change order t, the previous populations of MOEA solutions $P(t_o \leq k < t)$ become inappropriate since these solutions have lesser number of genes/tasks than what the new state of the system requires. We resolve this problem by inserting the new genes, corresponding to new tasks, to their immediate predecessors for all chromosomes in $P(t_o \leq k < t)$, as demonstrated in Figure 2 where tasks 31 to 40 are the new tasks. This gene insertion is only performed when there is an increase in task number. Further, the order of precedence for new tasks are preserved as demonstrated in Figure 2 where tasks 39 is preceded by task 38 in accord with Figure 1(b).

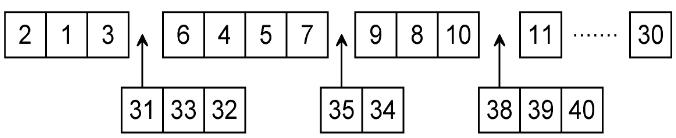

Fig. 2. Sample gene insertion

Considering that increases in number of tasks cannot be anticipated in practice, it is convenient to label genotype's task IDs from one to current number of tasks. With this setup, when task number increases, newly inserted tasks must have task IDs greater than the maximum ID of tasks in the previous environmental changes. As exemplified in Figure 2, this causes a large discontinuity of task ID values along the chromosome, which degrades CBA's performance as proven in [14].

Let the order of precedence of a task be defined as the maximum number of arcs that connects it to the start of the task precedence graph. For example, task 16 in Figure 1 (a) has 5 maximum number of arcs that connects it to the start of the precedence graph thereby making it to be of 5^{th} order of precedence. Now, the insertion side-effects was resolved in [14] by mapping IDs such that resulting IDs belonging to similar task order of precedence are as near as possible to each other. To illustrate, in Figure 1 (b), tasks at second order of precedence such as 5, 10, 14, 6, 7, and 12 where mapped to itself while tasks 31, 32, and 33 were mapped to 8, 9, and 11 respectively. Let the function $\mathcal{I}(d)$, where d is the original ID, represents this mapping operation. This function, as demonstrated in [14], resolves the insertion side-effect to the extent that it almost nullified the effect of increase in task number.

After the task insertion, the mapping function $\mathcal{I}(d)$ is applied to all elements in $P(t_o \leq k < t)$ then for each $k,\ t_o \leq k < t$, the preliminary centroid of this modified $P(k)$ is computed using Equation 4. Once again, this centroid could be infeasible. Now, for a centroid to represent its associated population as close as possible after being made feasible, it must be perturbed minimally. With this purpose, the function $\mathcal{R}$ in Equation 5 was modified to inflect minimal perturbation to the preliminary centroid as follows: Consider an already-made feasible section of the centroid comprised with genes having index $1 \leq k < i$ where G_i is the current gene under consideration. Now if G_i will make the section $1 \leq k \leq i$ feasible then its ID ID_i will not be replaced. Otherwise, it will be replaced by another gene not found in the feasible section which can make this section feasible and whose ID is nearest to ID_i. For example, suppose the first two genes of the centroid are made feasible already and that gene G_3, with ID of 3, does not make the first to third genes feasible. Further, suppose there are unassigned genes whose IDs are 1, 4, and 7 which are feasibly suitable replacements for G_3. Then gene of ID number 4 (closest to ID number 3) will replace G_3. Let the function $\mathcal{M}(G_i)$, called Minimal Repairer, represents this minimal perturbation that transforms infeasibility-causing gene to a feasibly suitable one. $\mathcal{M}(G_i)$ will replace $\mathcal{R}(G_i^p(t))$ in Equation 5 appropriately to formally define the minimal repair of the preliminary centroid.

A resource in our system carries ID of task utilizing it, such that the mapping operation is not only applied to gene's task ID but also to resource's task ID for the entire system to be in a mapped mode. Under this mode, MOEA computes the optimal solution for the t^{th} change producing population $P(t)$.

If the first increase in task number occurs at change order t and if there is no increase in task number at order $t+1$, the only steps involve are (1) the computation and (2) repair of centroid, and (3) feeding them to MOEA to compute for the optimal solution $P(t+1)$. The mapping of elements in $P(t_o \leq k < t+1)$ is not applied. If the second increase in task number occurs at $t+n_x$, the sequence of steps (1) – (3) are repeated up to $t+n_x-1$. At change order $t+n_x$, each $P(t_o \leq k < t+n_x)$ have all of its elements' gene ID inverse mapped $\mathcal{I}^{-1}$ and also the resource's task ID, i.e., the system will be restored to unmapped mode. Then the new tasks are inserted followed by a different mapping $\mathcal{I}$ – appropriate for the current state of the system – on gene ID of elements of $P(t_o \leq k < t+n_x)$. Again centroid of $P(t_o \leq k < t+n_x)$ is computed then repaired. Resource's task IDs are similarly mapped to restore the entire system back to mapped mode on which MOEA computes optimal solution $P(t+n_x)$. These schemes are applied appropriately to all succeeding changes. In this way, MOEA always sees a mapped system starting at the moment of first increase in task number. This approach is a vital partner of the mapping operation $\mathcal{M}$, since it facilitates the computation of optimal solutions – as environment changes by – to proceed as though task insertion has no effect.

Copy of the computed schedules/plans under mapped mode will be inverse-mapped when presented to decision-makers or when other parameters (such as set coverage or hypervolume) are needed to be extracted from them.

The Minimal Repairer $\mathcal{M}$ replacing $\mathcal{R}$ in Equation 5; the mapping operation $\mathcal{I}$; the insertion of new genes to all chromosomes; and the maintenance of system at mapped mode for MOEA; and the utilization of random immigrants in Equation 6 is called, Mapping Task IDs for Centroid-Based Adaptation with Random Immigrants (McBAR). All these innovations to CBA were demonstrated in [14] to enhance the search for optimal solution better than CBA; hence justifying them.

As in the case of CBA, McBA will not be employed to find the initial population that will be feed to MOEA to compute for baseline solution. Further, prior to an increase in task number, McBA is practically equivalent to CBA except for $\mathcal{M}$.

VII. Convergence Model

In the domain of Multi-Objective Optimization, it is of interest to find the hypervolume of NDS [26] which is related to the measure of optimality of NDS. Computation of hypervolume is based on minimization of objectives such that higher hypervolume values implies lesser objective values.

MOEA begins with an initial population whose individuals are located in some hypersurface with some hypervolume. As MOEA evolves this population, the evolved individuals could be at different points in the new hypersurface resulting to the population's hypervolume changing with respect to number evolution generation t. It would be of importance to find how fast hypervolume converge to a certain asymptotic value. To achieve this, we model hypervolume data by,

$$y\left(t\right) = DC - \frac{e^{P(t)}}{t+1} \tag{7}$$

where DC is a constant and is derived from the asymptotic horizontal line of the data on hypervolume; $P\left(t\right) = \sum_{n=0}^{N} c_n t^n$ is a polynomial whose coefficients are c_n and N chosen to avoid under and over fitting. The polynomial $P\left(t\right)$ makes $y\left(t\right)$ fits best given data of hypervolumes $d\left(t\right)$ in the least square sense, i.e.,

$$\min_{c_n} \left\| ln\left|\left(DC - d\left(t\right)\right)\left(t+1\right)\right| - P\left(t\right)\right\|^2 \tag{8}$$

The convergence rate is defined as the number of generations at which hypervolume is very close to its asymptotic value:

$$\tau = y^{-1}\left(DC - e^{c_0}\left[1-\xi\right]\right) \tag{9}$$

where ξ is a chosen value.

VIII. Experiments

Experiments were performed for RCPS problem type to show the behavior of McBAR, under proactive-reactive approach, with respect to environmental changes. As discussed in Section II, in revising schedules, on-going and finished tasks must be preserved to avoid high cost. Thus, McBAR is use to find an optimal revise schedule for the tasks which are not executed as yet only. However, this search is still based on cost, makespan, and probability of failure such that risk anticipation is still applied every time environment changes.

Types of environmental changes in these experiments are listed in Table I. The first column is for labels of change type, the second is for parameters that changes simultaneously. These types of changes are chained to form various sequences of changes, labeled by S_i in Table II. For example, sequence S_3 begins and is followed by a change in task duration (zero in Table I), then by simultaneous changes in task duration, task number, and resource availability (6 in Table I). The last column in Table II is the time at which changes occur. Within the sequence S_i is a set of task number increases. For example, S_1 has change types 2, 6, 5, 3 occurring at change orders 4, 6, 10, and 11 respectively where all of these change types involve task number changes based on Table I. The series of task number increases forms a sequence, labeled by T_k in Table III. For example, on S_1, task number increases sequence (TNIS) could be 3, 2, 2, and 3 at change orders 4, 6, 10, and 11, respectively. Various TNIS employed in the experiments are shown in Table III. There are 15 experiments types performed whose attributes are shown in Table IV. In particular, experiment labeled 7 is of sequence type S_1 and with TNIS T_5. Note that experiments at similar row has similar TNIS. Lastly, the experiment and change order pair $(X_\#, C_\#)$ will be called case.

TABLE I
Types Of environmental changes

Type	Parameter
0	task duration
1	resource availability
2	task number
3	task duration
	task number
4	task duration
	resource availability
5	task number
	resource availability
6	task duration
	task number
	resource availability

TABLE II
Sequences of changes of various types

Order \ Label	S_1	S_2	S_3	Time
1	0	0	0	4
2	0	0	0	6
3	0	2	6	8
4	2	1	0	12
5	0	0	4	13
6	6	3	0	16
7	1	0	0	19
8	4	4	5	23
9	0	6	1	26
10	5	5	2	30
11	3	0	0	33
12	0	0	3	37

A. Parameter Settings

Each experiment in Table IV has 30 starting tasks having precedence relationship given in Figure 1 when circled tasks removed. Being obliged to adapt to new situation during the execution of an RCPS plan, a total of 10 more tasks were

TABLE III
TYPES OF TASK NUMBER INCREASE SEQUENCE (TNIS)

Order \ Label	T_3	T_4	T_5	T_6	T_7
1	3	4	5	6	7
2	2	4	3	2	1
3	2	1	1	1	1
4	3	1	1	1	1

TABLE IV
TYPES OF EXPERIMENTS

S_1	S_2	S_3	TNIS
1	2	3	T_3
4	5	6	T_4
7	8	9	T_5
10	11	12	T_6
13	14	15	T_7

added, enclosed by circle. RCPS is applied to military mission planning and with this, the types of resources R_i are:

- Light Mortar Batteries (R_1) 16
- Infantry Companies (R_2) 17
- C130s (R_3) 5
- Apache helicopters (R_4) 16

The associated number to R_i is the limit on the total number of in-use resources.

During task duration change, task duration d_i is randomly varied following normal distribution N_m according to,

$$d_i \longleftarrow N_m(d_i + \delta, \delta) \tag{10}$$

with $\delta = 3.0$.

Our MOEA employs a population size of 40, mutation rate of .15, and crossover rate of .9, and terminates evolution after 2000 generations. To eliminate the stochastic property in MOEA, we repeated each experiment 30 times with different random seed. To compute for the probability of risk, our Monte Carlo simulation has 1000 simulations with task duration varied according to normal distribution having a standard deviation of 1.0. Note that variation of task duration for Monte Carlo simulation adds to that of task duration change by the environment. Further, when task duration is varied in the simulation the task starting times are fixed. Finally, we choose $\xi = .95$ in Equation 9.

IX. RESULTS AND DISCUSSIONS

Relationship of resource cost and makespan found through McBAR, under dynamic environment, was thoroughly discussed in our previous works [13], [14]. Probability of plan failure in relation to these two objectives will be discuss in this section with respect to environmental dynamics, types of environmental changes, and diversity of solutions to RCPS problem. Further, McBAR will be compared to an EA technique called Random Immigrant (RI).

A. Dynamics of Instances

As indicated in Section VIII, experiments are repeated for 30 times with each run having different random seed. Results from these runs at similar case (experiment-change order pair) were combined and plotted, called Risk Probability Distribution (RPD). Selected RPD are shown in Figure 3 with vertical axis as the resource cost, horizontal axis as the makespan, and grey-scale (yellow to blue scale in colored print) as the probability of plan failure; will be called risk value from here onwards. Risk values on identical cost-makespan point were averaged then interpolated using TriScatteredInterp command of Matlab. Subfigure labels correspond to case, e.g., Figure 3(1,5) is for results on various runs at case (1,5). This case does not have high risk – risk greater than .6; Figure 3(6,4) has high risk along the left boundary; and Figure 3(5,12) has high risk on the bottom boundary. RPD of distribution profile similar to Figures 3 (1,5), (6,4), and (5,12) are called, mild, left, and bottom, respectively. Not shown due to space limitation, some cases have RPD of high risk either in the middle, lower left, scattered or arcing.

TABLE V
RISK DYNAMICS ON VARIOUS SEQUENCES

Order	DRPD	S_1	DRPD	S_2	DRPD	S_3
1	mild	0	mild	0	mild	0
2	mild	0	mild	0	mild	0
3	mild	0	left	2	left	6
4	mild	2	left	1	various	0
5	various	0	mild	0	left	4
6	left	6	left	3	left	0
7	various	1	mild	0	various	0
8	mild	4	various	4	mild	5
9	mild	0	mild	6	various	1
10	left	5	left	5	various	2
11	left	3	left	0	left	0
12	left	0	left	0	left	3

When a makespan in RCPS is shorter, likely, more number of tasks were performed simultaneously at some time intervals and hence more resources are utilized simultaneously as well. Now if task durations are varied during Monte Carlo simulation – with task start times fixed – intuitively, there will be increase in number of resources consume simultaneously and hence more likelihood of defying resource constraint and therefore high probability of plan failure. This could explain the left type RPD in Figure 3(6,4). The logic being: the shorter makespan is or the less resources are invested the higher is the risk of failure in executing a plan. Figure 3(5,12) demonstrates high risk at lesser resource costs.

It was observed from the experimental results that cases (4,4), (7,4), (10,4) and (13,4) are all of mild type. These cases are for experiments 4, 7, 10 and 13, respectively, which all belong to sequence type S_1 and whose TNIS are T_4, T_7, T_{10}, and T_{13}, respectively, based on Table IV. In general, RPD was observed not to be influenced by TNIS type; a result that could be attributed to the efficiency of McBAR where change in number of task is almost nullified, as supported in [14]. For this reason DRPDs in Table V were arranged with respect to sequence type and change order only. The column labeled S_i is for sequence type whose entries are change types corresponding to the change order at the first column. Let

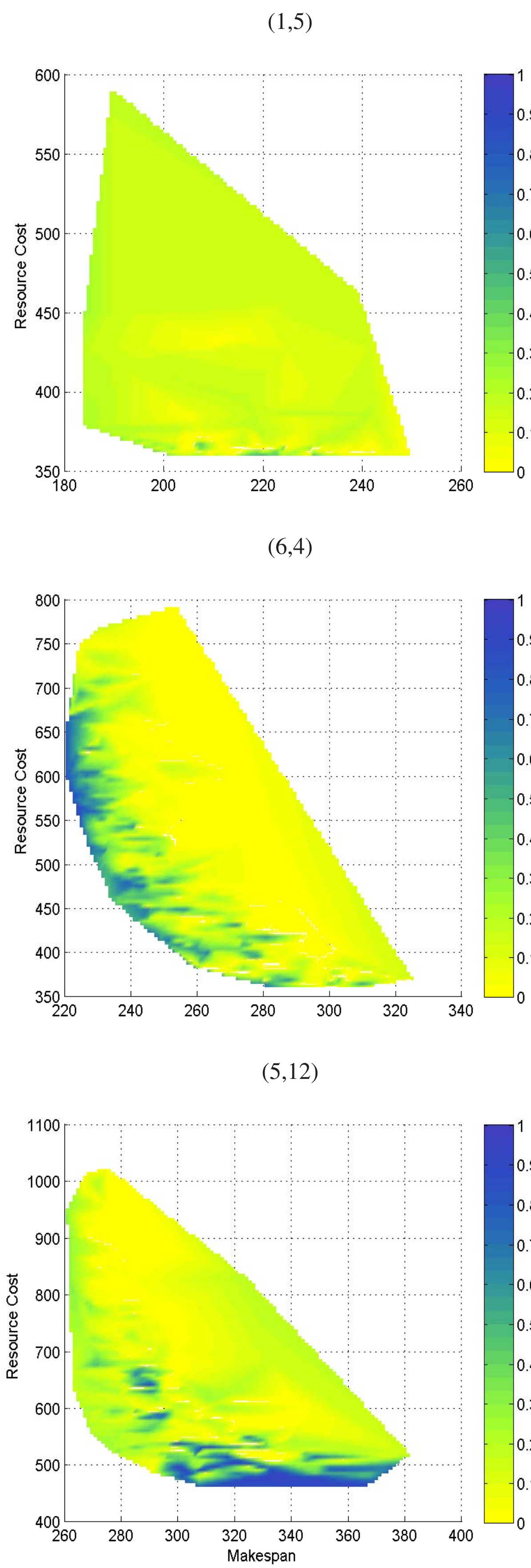

Fig. 3. Risk probability for selected cases (experiment, order)

sequence-change order pair be called instance. Note that some cases of similar sequence type does not have dominant RPD types. The entry for this in the table is “various”, such as instance (8,4).

In Table V, transitions from change type zero (change in task duration) to type zero – such as instances (2,11) to (2,12), (1,2) to (1,3), etc. – did not change DRPD in general. This could imply that successive random changes in task duration is not compounding. Also, DRPD did not change in the transition between change types 2 and 1. Type 6 in Table I is a simultaneous changes in task duration, task number, and resources. Transitions between Types 6 and 0 in Table V – such as instances (1,5) to (1,6), (3,2) to (3,3), etc. – always result to DRPD of left RPD type which could be due to the simultaneity in Type 6 change. Same could be said in transition between Types 3 and 0.

Seemingly random changes in DRPD on other transitions could be explained by the non-linearities of the system.

B. Types Of Changes

Relationship of the three objectives will be illustrated in this section with respect to change type. As above, results relating the three objectives from all runs are combined with risk values at similar cost-makespan location averaged then the resulting set of risk values interpolated. Let the resulting plot of the three objectives be called RPD per Change (RPDC). Not shown, RPDC for change type zero shows high risk values in region of lesser to average makespan. The high risk values at lesser makespan conforms with the stated logic above while those at average makespan could be attributed to various transitions between change type zero and other change types. As cited above, change in task number is almost nullified by McBAR. Consequently, change type 3 – simultaneous change in task duration and task number – could effectively be type zero (change in task duration only); as supported by the observation that type 3 RPDC is strongly related to that of type zero. RPDC for change type 4 – a simultaneous change in resources and task duration – shows higher risk values at lesser makespan, which conforms with the logic above, but at larger range and higher values of resource cost, which could be attributed to transitions between this type and other change types.

RPDC for other change types have sparse region of high risks which could be attributed to system’s non-linearity.

C. Hypervolume Dynamics

The dynamics of hypervolume with respect to evolution generation looks like those of Figure 4 where it starts at lower values and levels off starting at 1000 generations. The DC level in Equation 7 is the asymptote of this dynamics. For similar reason associated with Table V, DC values in Table VI are the averages over runs and over cases of similar TNIS. Further, Table VI has similar structure to that of Table V except for DRPD replaced by DC levels.

The DC level between the change type transitions zero and 2 – such as instances (1,3) to (1,4), (1,4) to (1,5), etc. – are approximately equal which could be attribute to the

TABLE VI
HYPERVOLUME DC LEVELS ON ALL SEQUENCES TYPES

Order	DC Level	S_1	DC Level	S_2	DC Level	S_3
1	911,629	0	919,095	0	905,213	0
2	941,764	0	935,753	0	945,411	0
3	941,058	0	932,088	0	916,516	6
4	939,240	2	918,146	2	895,056	0
5	943,035	0	900,549	0	911,868	4
6	896,815	6	901,722	6	907,495	0
7	931,317	1	910,256	1	926,900	0
8	898,375	4	930,372	4	899,591	5
9	900,074	0	902,876	0	909,605	1
10	915,373	5	912,832	5	911,497	2
11	914,559	3	907,152	3	913,815	0
12	911,280	0	911,127	0	911,081	3

effectiveness of McBA where change type 2 is effectively type zero. The change type transition from zero to 6 – such as in (1,5) to (1,6), (3,2) to (3,3), etc. – is an increase in complexity considering that type 6 has changes in both task duration and resources while type zero only have task duration change. This increase in complexity could degrade McBAR's performance in its search of optimal solution, as supported in our previous work [21], and hence lower objective values can hardly be reached resulting to lower DC levels. The converse happens to the transition from change type 6 to zero, 5 to one, and 6 to one where there is a decrease in complexity. The transitions from change type 5 to 3, zero to 3, 3 to zero, and one to 2, by the efficiency of McBAR, are effectively transitions from change type one to zero, zero to zero, zero to zero, and one to zero, respectively. Thus, no significant change in problem complexity and by the previous reasoning will result to approximately equal DC level as reflected in the table.

All unaccounted transitions cannot be generalized and could be due to system nonlinearities.

D. Diversity on Types of Changes

The hypervolumes computed for a given change type were averaged. Selected results are plotted in Figure 4 where the noisy graph is the actual data and the smooth graph as the fitted data derived through Equation 7. The "*" in the figure is the point whose generation component is the convergence rate τ in Equation 9. For change types zero to 6 the convergence rates are 146, 279, 135, 143, 105, 555 and 163 respectively. As supported in [21], complex changes degrades McBAR's performance which will cause it to take more generations to find optimal values than for simpler types of changes. It is then expected that complex change types require larger converge rate. The enumerated convergence rates increases from change type zero to one, and from four to five which could be explain by the increase in complexity. As previously mentioned, McBAR's efficiency almost nullify the effect of change in task number, which could explain the difference in convergence rate from type one to two where type two is effectively no change at all. Same reasoning applies to explain the difference between type two – effectively no change – and type 3 – change in duration effectively.

Unexpectedly, convergence rate for change type 4 is lesser than that of type zero; a change of lesser complexity. Same could be said between change type 6 and 2, and 6 and 5. These results could be due to system non-linearities.

E. McBAR compared to RI

To manifest further the versatility of McBAR in dealing with problems dealt in this paper, we compare this with another technique called Random Immigrant (RI). This latter technique produces a set of chromosomes randomly guided only by Serial Schedule Generation Scheme (SSGS) [27] for the chromosomes to be task-precedence feasible. By this randomness, comparing RI with McBAR could emphasize well the relevance of McBAR's methodology.

We perform all experiments mentioned above but this time employing RI instead of McBAR and compare the output of these two techniques using set coverage, defined in Section IV. Results of this comparison showed the superiority of McBAR over RI in almost all cases. However, there are very rare cases at which RI is superior over McBAR. Table VII shows the cases at which RI set coverage is greater than .1, i.e., $SC\left(RI, McBAR\right) - SC\left(McBAR, RI\right) > .1$. Most of these cases has change types either zero (task duration change) or 4 (simultaneous change in task duration and resource availability), cases at which there are no changes in task number where McBAR could be expected not to perform best considering that it is design foremost to combat change in task number. Cases (1,10) and (3,8) has change type 5 which is a simultaneous change in task duration and number. This could be explain by the result of our previous work [21] that showed the strong influence of simultaneity of change on the degradation of McBAR's performance.

TABLE VII
SELECTED SET COVERAGE

Experiment	Order	Technique	RI	McBAR
1	8	RI	N/A	0.53 ± 0.06
		McBAR	0.19 ± 0.04	N/A
	9	RI	N/A	0.54 ± 0.05
		McBAR	0.14 ± 0.04	N/A
	10	RI	N/A	0.46 ± 0.05
		McBAR	0.17 ± 0.04	N/A
3	8	RI	N/A	0.39 ± 0.07
		McBAR	0.24 ± 0.05	N/A
	10	RI	N/A	0.38 ± 0.05
		McBAR	0.22 ± 0.04	N/A
	11	RI	N/A	0.41 ± 0.06
		McBAR	0.13 ± 0.03	N/A
8	8	RI	N/A	0.35 ± 0.05
		McBAR	0.23 ± 0.05	N/A
15	5	RI	N/A	0.49 ± 0.07
		McBAR	0.27 ± 0.05	N/A
	6	RI	N/A	0.43 ± 0.06
		McBAR	0.24 ± 0.05	N/A
26	8	RI	N/A	0.50 ± 0.07
		McBAR	0.19 ± 0.05	N/A
27	5	RI	N/A	0.58 ± 0.06
		McBAR	0.21 ± 0.05	N/A
	6	RI	N/A	0.49 ± 0.06
		McBAR	0.22 ± 0.05	N/A
	7	RI	N/A	0.52 ± 0.06
		McBAR	0.19 ± 0.05	N/A

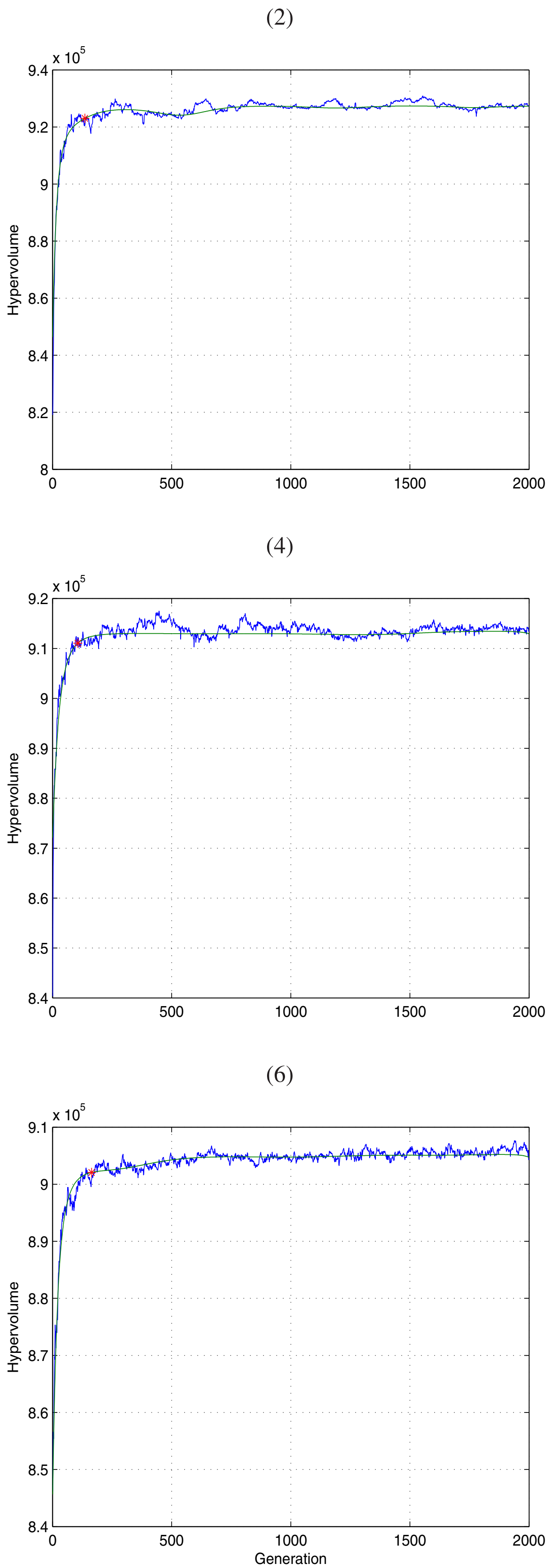

Fig. 4. Hypervolume as a function of generation

X. Conclusion and Future Works

Our previous works showed the efficiency of a memory-based approach called, Mapping of Task IDs for Centroid-Based Adaptation with Random Immigrants (McBAR), in searching for optimal solution on dynamic problem. This paper applies McBAR to dynamic Resource-Constraint Project Scheduling (RCPS) reactively and proactively having resource cost, makespan, and risk of failing to implement schedule as objectives under an environment that could change task duration, resource availability, and number of tasks. It is our goal to show how McBAR's efficiency behaves when there are three conflicting objectives under dynamic environment and to perform risk analysis on dynamic RCPS. Our experimental results showed that McBAR maintained its efficiency despite the increase of objective number. Further, they showed risk to be higher when there are lower schedule makespan and/or lower resources invested to a schedule/plan. Our other goal was to analyze optimization-related parameters such as hypervolume. Results showed that, in general, hypervolume converges to lower/higher values when environment changes to more/less complex types of changes, with few unexpected results. Lastly, McBAR performs superbly compared to another Evolutionary Computation approach called, Random Immigrants.

We plan to investigate in the future the role of Evolutionary Algorithm parameters, such as crossover and mutation rate, on behaviors of McBAR and optimization-related parameters under problem such as presented in this paper. Further, we plan to compare those behaviors between problem with or without risk under dynamic environment.

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] O. Lambrechts, E. Demeulemeester, and W. Herroelen, "Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities," *Journal of Scheduling*, vol. 11, no. 2, pp. 121–136, 2007.

[2] L. Bui, M. Barlow, and H. Abbass, "A multiobjective risk-based framework for mission capability planning," *New Mathematics and Natural Computation*, vol. 5, no. 2, pp. 459 – 485, 2009.

[3] B. Mo, A. Gjelsvik, and A. Grundt, "Integrated risk management of hydro power scheduling and contract management," *IEEE Transactions on Power Systems*, vol. 16, pp. 216–221, 2001.

[4] M. Bruni, P. Beraldi, F. Guerriero, and E. Pinto, "A heuristic approach for resource constrained project scheduling with uncertain activity durations." *Computers & OR*, vol. 38, no. 9, pp. 1305–1318, 2011.

[5] R. Hochreiter, "Evolutionary multi-stage financial scenario tree generation." in *EvoApplications (2)*, ser. Lecture Notes in Computer Science, C. D. Chio, A. Brabazon, G. A. D. Caro, M. Ebner, M. Farooq, A. Fink, J. Grahl, G. Greenfield, P. Machado, M. O'Neill, E. Tarantino, and N. Urquhart, Eds., vol. 6025. Springer, 2010, pp. 182–191.

[6] T. Tometzki and S. Engell, "Risk management in production planning under uncertainty by multi-objective hybrid evolutionary algorithms," in *20th European Symposium on Computer Aided Process Engineering*, ser. Computer-Aided Chemical Engineering, S. Pierucci and B. Ferraris, Eds., vol. 28, 2010, pp. 151–156.
[7] B. Roorda, "An algorithm for sequential tail value at risk for path-independent payoffs in a binomial tree." *Annals OR*, vol. 181, no. 1, pp. 463–483, 2010.
[8] Y. Yang and J. Lee, "Probabilistic modeling and dynamic optimization for performance improvement and risk management of plant-wide operation." *Computers & Chemical Engineering*, vol. 34, no. 4, pp. 567–579, 2010.
[9] T. Ouarda and J. Labadie, "Chance-constrained optimal control for multireservoir system optimization and risk analysis," *Stochastic Environmental Research and Risk Assessment*, vol. 15, pp. 185–204, 2001.
[10] S. Elloumi and P. Fortemps, "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem," *European Journal Of Operational Research*, vol. 205, no. 1, pp. 31–41, August 16 2010.
[11] M. Hapke, A. Jaszkiewicz, and R. Slowinski, "Fuzzy multi-mode resource-constrained project scheduling with multiple objectives," in *Recent Advances in Project Scheduling*, J. Weglarz, Ed. Kluwer Academic Publishers, 1998, ch. 16, pp. 355–382.
[12] F. Ballestin and R. Blanco, "Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems," *Computers and Operations Research*, vol. 38, no. 1, pp. 51–62, January 2011.
[13] 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.
[14] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An Adaptive Approach for Solving Dynamic Scheduling with Time-varying Number of Tasks – Part I," in *Proceedings of the 2011 IEEE Congress on Evolutionary Computation*, 2011.
[15] 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.
[16] J. Branke, *Evolutionary optimization in dynamic environments*. Massachusetts USA: Kluwer Academic Publishers, 2002.
[17] S. Van de Vonder, E. Demeulemeester, and W. Herroelen, "Proactive heuristic procedures for robust project scheduling: an experimental analysis," *European Journal of Operational Research*, vol. 189, no. 3, pp. 723–733, 2008.
[18] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
[19] Y. Jin and B. Sendhoff, "Trade-off between performance and robustness: an evolutionary multiobjective approach," in *EMO'03: Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization*. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 237–251.
[20] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *J. Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.
[21] 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 2011 IEEE Congress on Evolutionary Computation*, 2011.
[22] K. Willick, S. Wesolkowski, and M. Mazurek, "Multiobjective evolutionary algorithm with risk minimization applied to a fleet mix problem," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–7.
[23] P. Artzner, F. Delbaen, J. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.
[24] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. New York: John Wiley and Son Ltd, 2001.
[25] 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.
[26] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[27] S. Hartmann and R. Kolisch, "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 127, no. 2, pp. 394–407, 2000.