2013 IEEE International Conference on Systems, Man, and Cybernetics

Using Genetic Algorithms in Effects-based Planning

Irfan Younas and Rassul Ayani
School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden
irfany@kth.se, ayani@kth.se

Johan Schubert and Hirad Asadi
Department of Decision Support Systems
Division of Information and Aeronautical Systems
Swedish Defence Research Agency
SE-164 90 Stockholm, Sweden
johan.schubert@foi.se, hirad.asadi@foi.se

***Abstract*—In this paper, we propose a genetic algorithm-based method for evaluation of operational plans within effects-based planning. We formulate the effects-based planning problem as a bi-objective optimization problem, in which the distance from the initial state to the current state (g) and the distance from the current state to the desired end state (h) are minimized. To solve the problem, we adopt Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Considering an expeditionary operation scenario, we simulate a subset of possible plans and present the decision maker with a set of promising plans which are capable of approaching the desired end state efficiently. In order to discuss the efficiency and effectiveness of the algorithm, we compare the results of NSGA-II with the results of A*. The computational results show that NSGA-II is much more efficient than A* with regard to g. On the other hand A* is a little more effective with regard to h.**

Keywords*—*Effects-based Planning, Genetic Algorithms, Search Algorithms, Path Planning, Optimization

I. Introduction

How we model a phenomenon depends on the purpose of the model and the questions we want to answer. Since our simulation system aims to support decision-making within an effects-based approach to operations (EBAO) [1][2] the modeling has to be based on EBAO and the concepts used within it, such as plan, action, effect, and end state.

EBAO is a military approach to the management and implementation of efforts at the operational level. EBAO is defined in [3] as: "operations that are planned, executed, assessed, and adapted based on a holistic understanding of the operational environment in order to influence or change system behavior or capabilities using the integrated application of selected instruments of power to achieve directed policy aims". Within the framework of EBAO, effects-based planning (EBP) is a method for developing objectives and effects to be achieved through a series of synchronized actions within a military operational plan, conceptually developed starting top-down from a desired end state.

A. Problem Formulation

We make use of the same scenario that has regularly been used by the Swedish Armed Forces in the Combined Joint Staff Exercises. In Bogaland, a newly industrialized country, a civil war broke out ten years ago when discontent within the minority ethnic-religious group had reached very high levels. The incumbent presidents have signed a peace-agreement, and an international force, BFOR, is present to support the implementation of the agreement.

With a decision support tool developers of operational plans are able to evaluate thousands of alternative plans against possible courses of events and decide which of these plans are capable of achieving a desired end state. The purpose is to understand the consequences of different plans through simulation and evaluation. Operational plans are described in the effects-based approach to operations concept as a set of actions and effects. For each action we may have several different alternative ways to perform the action. Together they make up all possible plans, which are represented as a tree of action alternatives that may be searched for the most effective action alternative sequences. The task of the planner is to find a set of effective, efficient and robust plans. These plans need to be effective in reaching the end state or coming close to the end state with minimal effort, and should preferably be in a neighborhood of other similar plans [4][5]. Thus our goal is to minimize both the distance of the current state to the desired end state, denoted by h, and overall consequence of all performed actions as a distance from the initial state to the current state, denoted by g. Thus, this is a multi-objective optimization problem.

Schubert et al. [5] describe simulation-based decision support techniques for evaluation of operational plans within effects-based planning. The distance function f, which is the sum of g and h, is minimized by A*. As the functions g and h are conflicting, instead of treating them as a single objective, in this paper we formulate the planning problem as a bi-objective optimization problem and solve it using multi-objective genetic algorithms. Many researchers [6][7][8][9] have used genetics algorithms for solving multi-objective optimization problems. However, to the best of our knowledge, the existing work does not address the multi-objective optimization model of the planning problem discussed in this paper.

B. Simulation System

We have developed a simulator that enables a military analyst to identify the best military plans among a large number of possible action alternatives [10]. The general idea is that when there are many possible combinations of actions for a military plan then a simulator can evaluate these combinations in order to show the decision maker which possible combinations are most successful.

The simulator is based on discrete events and the simulation engine uses A* algorithm (described in the next section)

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.80

to find the best action combinations.

II. The Proposed Algorithm

The planning problem discussed in the previous section is a very large combinatorial optimization problem. Due to large search space, an enumerative algorithm may take many years to solve the problem. Evolutionary Algorithms (EAs) are well-known metaheuristics for sampling intractably large and highly complex search spaces. In this paper, we formulate the planning problem as a bi-objective optimization problem and solve it using Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [11]. NSGA-II is a well-known evolutionary algorithm to solve multi-objective optimization problems. A* can also be used for searching the optimal plan. A* is an algorithm for finding lowest expected total distance from initial node to one goal node. In subsection III-E, we also compare the results of NSGA-II and A*. We use the NSGA-II implementation provided by jMetal framework [12] with some modifications to avoid duplicate solutions (plans). In the next two subsections, we describe the steps and parameter settings of NSGA-II and A* for the proposed problem.

A. NSGA-II Algorithm

In order to use Genetic algorithms to solve an optimization problem, the first step is to devise a suitable representation scheme. We consider a scheme in which a plan (chromosome in GA) is represented by a n-dimensional vector of main activities and a sequence of sub-activities within main activities. An example chromosome for a problem shown in Fig. 1 consists of eight main activities, denoted by A_1 to A_8, and their sub-activities. We can also refer to main activities and sub-activities as "actions". For each action, we may have several different alternative ways to perform it. The tree of these action alternatives represents all possible plans. In Fig. 1, numbers 1, 2, 41, 52, 54, 61, 78, and 47 denote Ids of the chosen alternatives of the eight main activities. Each of the main activity has zero or more sub-activities. For instance the main activity 41 has one sub-activity and Id for the selected alternative of the sub-activity is 17. The overall action alternatives sequence is shown in Fig. 2, where the sample plan comprises of 16 actions in total, 8 main activities marked in grey and 8 sub-activities.

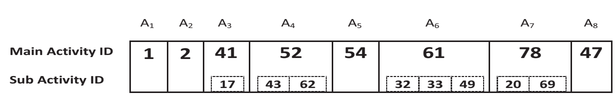

Fig. 1. Chromosome representation of a candidate solution (Plan)

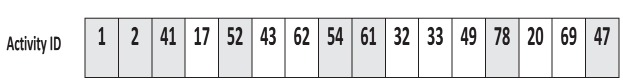

Fig. 2. A plan consisting of a sequence of actions

The NSGA-II algorithm can be summarized:

- Initialize population P_0 with N non-duplicate candidate solutions (plans) chosen uniformly at random. Each plan is a sequence of actions.
- Each chosen plan is evaluated using our simulator to find its fitness.
- For $t = 0$ to $M - 1$ (M is the total number of generations) do:
  - Generate population Q_t comprising of N non-duplicate offsprings. The offspring generation process undergoes the following three procedures.
    Selection: In order to generate an offspring, two parent solutions are selected using binary tournament.
    Crossover: Using one-point crossover operator, the chosen parents are combined to generate an offspring as shown in Fig. 3. The crossover point c_p is chosen uniformly at random. The offspring inherits first c_p main activities (genes) from parent A, while the remaining genes are taken from parent B.

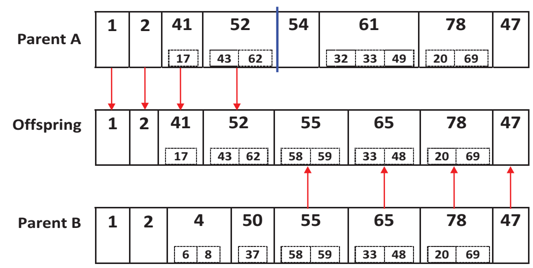

Fig. 3. One-point crossover operation

    Mutation: In mutation, we alter values of one or more activities (genes) of the constructed plan (child chromosome). The mutation probability for each main activity is calculated as $m_p = 2.0/n$, where n is the number of main activities. For each main activity A_i, where $i \in \{1, 2, \ldots n\}$, we draw a number d_i uniformly at random. If d_i is less than m_p, then we need to mutate A_i. We randomly choose one of the available alternatives for A_i and the corresponding sub-activities are also selected uniformly at random. For instance, two main activities (A_3 and A_6) are selected for mutation as highlighted in red in Fig. 4. First consider A_3, we randomly select one of the available alternatives, for instance alternative 45. We must also select sub-activity alternatives if there are any. In this example, we assume that the selected activity alternative 45 has no sub-activity. For A_6, suppose the selected alternative is same, i.e., 65. In this case, we randomly choose alternatives for each sub-activity, which are 34 and 46 in the mutated offspring.
  - All generated offsprings in Q_t are evaluated using our simulator.
  - Build the population U_t such that $U_t = P_t \cup Q_t$. The size of U_t is $2N$.
  - Rank all plans in U_t into non-dominated fronts according to Pareto-dominance relationship.

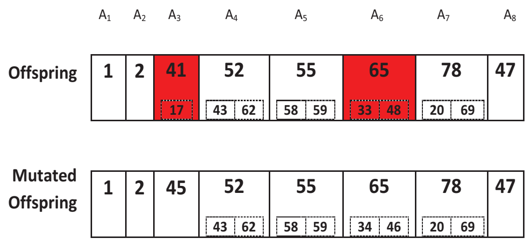

Fig. 4. Mutation Operator

- ◦ Next generation population P_{t+1} contains N plans, which are taken from the these non-dominated sets in sequence. While iterating over each front, if the size of the front is greater than the number of remaining solutions then crowding distance is used.
- Return the top non-dominated set, which is a Pareto-front approximated by NSGA-II.

B. *A*-search*

To find good combinations of alternatives for all actions of the plan we apply A*-search. It means that, on the basis of a given system state, we simulate the effect of each alternative action in our plan, but only one step at the time. Doing so, for every alternative, we get a new system state whose distance to the desired end state, denoted by h, is calculated. Given the alternative that is best, i.e., closest to our end state, we simulate possible subsequent alternative actions provided, but again only one step ahead in our action/event list. One of these alternatives leads to a condition that is closer than the others. However, it is possible that all the alternatives actually lead away from the target.

Therefore, we must also compare the new distance with the best of the distances that have been simulated and recorded in the previous simulation steps, but then had opted out in favor of a better sequence of alternative actions. The best sequence now becomes the basis for the next simulation step.

During simulation an assessment is made of how well each action is performed. This is done by the functions g and h. Function g measures the consequence of all performed actions as a distance from the initial state $S_{0,0}$ to the current simulated state S_{x,y_x} action-by-action [4][5].

Function h is a heuristic estimate of the remaining distance from S_{x,y_x} to the end state. The total weighted estimated distance from the initial state to the end state via the current state S_{x,y_x} is

$$f(y_x) = g(y_x) + 80h(y_x). \quad (1)$$

This is the distance function that is minimized by A*. The weight "80" was derived by experimentation to balance the performance of minimizing g and h and is domain dependent.

III. Experimental Results

In this section, a set of computational experiments and their results are presented. The aim of these experiments is to find those plans which are capable of achieving a desired end state. First, we describe the problem instance, the parameters of the algorithm, and the methodology considered in the experiments. After that we present the computational results of NSGA-II and the comparison of NSGA-II and A*.

We implemented our GA in Java and run it on a PC with an Intel Core $i5 - 2.60$ GHz and 4.0 GB of RAM. The implementation of the presented algorithm is based on Java-based framework jMetal [12] with some modifications for the discussed problem, crossover and mutation.

A. *EBP Problem Instance*

Consider an expeditionary operation where a plan consists of m actions and each action a_i has c_i choices. The plans affect p actors, where an actor is made of either one or more entities. Each actor is described by q parameters (features) and each parameter can have integer values between 0 and 3 inclusive. For our experiments, we assume that total number of actions vary from 34 to 42 since each "main" action can contain zero or more sub-actions. The total number of main actions is 28.

In the considered scenario, we assume 40 different actors ($p = 40$), where each actor is described by 15 parameters ($q = 15$). Each actor assumes some desired values for these parameters. Also, each performed action has some influence on different actors. This means that when an action has been executed, the parameters for specific actors, which are targeted by the action, are updated. For example, if an action is called "Neutralize Irregular Forces", this means that BFOR will target certain actors that are considered to be hostile among the 40 actors.

The actor parameters are used to calculate the distance values, namely the g and h values for each performed action. Since we are interested in optimal plans where fewer resources are consumed to reach the military end state, the objective is to minimize the g and h values.

B. *NSGA-II Parameters*

For experiments, the initial population size is set to 150. The algorithm terminates after 2000 plan evaluations. Mutation probability for each gene (main activity) is $2/28$. The alternatives for sub-activities are chosen uniformly at random. The outcome of an action can vary depending on the circumstances (the operation may even fail), which can be addressed by making the simulation stochastic [5]. Due to the stochastic nature, the simulator uses Monte Carlo simulations for obtaining a frequency function of the entire outcome space. In our experiments, we set the Monte Carlo simulation parameter equal to 20.

C. *Experimentation Methodology*

Due to stochastic nature of the problem, we run 20 replications for each evolved solution (plan) found by the algorithm and calculate mean of g and h, where g is the total distance covered from the start location to the current position and h is the estimated distance from the current position to the goal (end) state.

In subsection III-E, we compare the results of NSGA-II with A*. The methodology for the comparison is as follows. From 10 000 plans simulated using A*, we select the top 100 based on f value ($f = g + h$) and run 20 replications for each to compute $\overline{g}$ and $\overline{h}$. Similarly we run NSGA-II with the initial population size $N = 150$ and choose the top 100 from the evolved 150 plans based on f value. To compute $\overline{g}$ and $\overline{h}$, we run 20 replications for each. After calculating $\overline{g}$ and $\overline{h}$ for 100 plans, we compare the results of NSGA-II and A* in two separate graphs.

In multi-objective problem, instead of one optimal solution, we have a set of solutions. We have a Pareto-set which is a set of non-dominated solutions. For A* and NSGA-II, we find a set of non-dominated plans for each. In order to evaluate the obtained solutions of a multi-objective problem, we need to assess the quality of the obtained non-dominated set of solutions [6]. The quality of the obtained solutions depends on two properties: (i) convergence to the Pareto front, (ii) diversity of solutions. In order to measure these two properties, a number of quality measure indicators have been proposed [12]. Some of them are Hypervolume (HV) [7], Epsilon [13], Generational Distance [8], Inverse Generational Distance, and Spread [9]. HV is considered to be most widely used and an accurate indicator to measure both properties (convergence and diversity) at the same time [6].

The volume covered by the candidate solutions of the obtained Pareto front approximation in the objective space is calculated by HV. Fig. 5 illustrates a bi-objective problem where two objectives f_1 and f_2 should be minimized. The approximated Pareto consists of 5 solutions (S1, S2, S3, S4, and S5), whereas the true Pareto-optimal is unknown. HV of the obtained Pareto front is calculated with respect to the reference point R. HV is the union of the areas enclosed by the set of non-dominated solutions and R. This is shown in Fig. 5, where HV is the volume enclosed by the outer dotted border. The worst value of each objective from a Reference Pareto Front (RPF) is used for calculating a reference point. RPF is the set of all non-dominated solutions found in different experiment settings for a specific problem instance. The solutions in Pareto set are normalized between 0 and 1 to ensure that HV is always equal to or less than 1. In our problem the RPF is composed of all non-dominated solutions obtained by NSGA-II and A* algorithms.

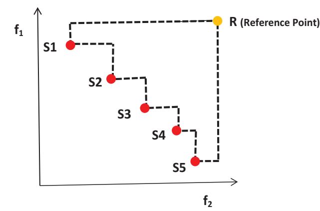

Fig. 5. Hypervolume of a non-dominated set of 5 solutions

D. Computational Results of NSGA-II

In this section we present the computational results of the planning problem which are obtained using NSGA-II. The results shown in this section correspond to initial population size $N = 150$ and 2000 plans evaluations (after 2000 evaluations we terminate). The objective is to minimize g and h. Here we present the decision maker with 100 such alternative plans which are most close to the desired end state with minimum effort. To make it simple, we select the top 100 based on f value from 150 evolved plans. We run 20 replications for each evolved selected plan and calculate mean values for g and h. Because of room constraints, we just present the top 10 plans based on $\overline{g}$ and $\overline{h}$ in Tables I and II.

Table I shows the best 10 alternative plans with respect to $\overline{g}$, where $\overline{g}$ of a plan is the average consequence of all performed actions as a distance from the initial state to the current state. Table II shows the best 10 alternative plans with respect to $\overline{h}$, where $\overline{h}$ of a plan is the estimated distance of the plan to the desired goal state. The smaller is the distance, the closer to the goal state we are. Decision makers can choose the plans considering both of these criteria $\overline{g}$ and $\overline{h}$. For example if a decision maker wants to minimize the total distance from the initial state to the current state then he/she should prefer the first plan with $\overline{g} = 2261.33$ and 34 activities (see Table I). The second plan in Table I also consists of 34 activities but it is worse than the first one in both criteria. There is only one activity in the second plan which is different from the first one and that is activity number 28 (103 instead of 102). Though both $\overline{g}$ and $\overline{h}$ for the second plan are worse than the first one, still the decision maker may be interested in such a plan as a backup.

On the other hand if a decision maker is interested in

TABLE I
TOP 10 PLANS IDENTIFIED BY NSGA-II BASED ON $\overline{g}$

#	Plan	Objective Values	Function
		$\overline{g}$	$\overline{h}$
1	{1, 2, 41, 42, 44, 6, 20, 7, 12, 62, 64, 50, 60, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 102, 106, 29, 30, 58, 46, 47}	2261.33	816.705
2	{1, 2, 41, 42, 44, 6, 20, 7, 12, 62, 64, 50, 60, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 103, 106, 29, 30, 58, 46, 47}	2266.495	818.03
3	{1, 2, 41, 61, 44, 6, 20, 7, 11, 65, 49, 50, 60, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 103, 106, 29, 30, 58, 46, 47}	2266.575	818.115
4	{1, 2, 41, 42, 43, 6, 20, 5, 12, 65, 49, 50, 60, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 102, 105, 29, 30, 57, 46, 47}	2268.065	817.635
5	{1, 2, 41, 42, 43, 6, 20, 5, 12, 62, 64, 50, 51, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 103, 107, 29, 30, 58, 46, 47}	2268.18	818.8
6	{1, 2, 41, 61, 43, 108, 20, 7, 12, 65, 49, 50, 51, 52, 17, 63, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 105, 29, 30, 58, 46, 47}	2268.465	816.765
7	{1, 2, 41, 61, 43, 6, 20, 7, 12, 4, 49, 50, 60, 52, 16, 3, 54, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 105, 29, 30, 57, 46, 47}	2268.475	817.675
8	{1, 2, 41, 42, 44, 108, 20, 5, 12, 4, 64, 50, 60, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 102, 106, 29, 30, 58, 46, 47}	2269.8	817.305
9	{1, 2, 41, 61, 43, 6, 20, 7, 12, 62, 64, 50, 51, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 103, 106, 29, 30, 58, 46, 47}	2270.385	818.06
10	{1, 2, 41, 42, 44, 108, 20, 7, 12, 62, 49, 50, 51, 52, 17, 63, 53, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 105, 29, 30, 57, 46, 47}	2270.635	817.825

TABLE II
TOP 10 PLANS IDENTIFIED BY NSGA-II BASED ON $\overline{h}$

#	Plan	Objective Values $\overline{g}$	Function $\overline{h}$
1	{1, 2, 41, 42, 44, 108, 20, 10, 23, 14, 109, 4, 64, 50, 51, 52, 16, 63, 54, 55, 56, 68, 75, 67, 86, 90, 79, 80, 94, 98, 95, 96, 102, 105, 31, 32, 33, 34, 57, 46, 47}	2708.35	814.95
2	{1, 2, 41, 42, 43, 6, 20, 5, 13, 15, 65, 49, 50, 60, 52, 16, 3, 53, 55, 56, 66, 67, 81, 93, 79, 80, 97, 95, 96, 104, 106, 31, 32, 33, 34, 57, 46, 47}	2509.7	815.42
3	{1, 2, 41, 42, 44, 6, 20, 7, 12, 62, 64, 50, 60, 52, 18, 28, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 106, 29, 30, 58, 46, 47}	2338.23	815.785
4	{1, 2, 41, 42, 44, 108, 20, 7, 12, 65, 49, 50, 51, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 107, 29, 30, 57, 46, 47}	2276.47	815.81
5	{1, 2, 41, 61, 43, 108, 20, 8, 23, 14, 15, 65, 64, 50, 60, 52, 16, 63, 48, 55, 56, 69, 75, 67, 85, 93, 79, 80, 94, 100, 95, 96, 102, 105, 29, 30, 58, 46, 47}	2584.255	815.895
6	{1, 2, 41, 42, 44, 108, 20, 5, 11, 4, 64, 50, 60, 52, 16, 3, 48, 55, 56, 72, 78, 67, 84, 92, 79, 80, 97, 95, 96, 103, 105, 31, 32, 33, 34, 58, 46, 47}	2516.075	816.14
7	{1, 2, 41, 42, 44, 6, 20, 7, 14, 15, 65, 49, 50, 60, 52, 16, 3, 53, 55, 56, 69, 77, 67, 84, 92, 79, 80, 97, 95, 96, 103, 105, 31, 32, 33, 34, 58, 46, 47}	2580.005	816.21
8	{1, 2, 41, 42, 43, 6, 20, 9, 24, 13, 15, 4, 64, 50, 60, 52, 16, 63, 53, 55, 56, 66, 67, 83, 92, 79, 80, 97, 95, 96, 104, 106, 31, 32, 33, 34, 57, 46, 47}	2575.23	816.295
9	{1, 2, 41, 61, 43, 6, 20, 7, 12, 4, 49, 50, 60, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 94, 100, 95, 96, 102, 105, 29, 30, 58, 46, 47}	2322.025	816.32
10	{1, 2, 41, 42, 44, 108, 20, 7, 11, 62, 49, 50, 51, 52, 17, 63, 53, 55, 56, 66, 67, 89, 79, 80, 97, 95, 96, 101, 105, 29, 30, 57, 46, 47}	2274.535	816.375

those plans which come closer to the goal state irrespective of the consequences of all performed actions then he/she should prefer the first plan with $\overline{h} = 814.95$ as shown in Table II. The plan consists of 41 activities and the total distance from the initial state to the current state is 2708.35. The $\overline{h}$ values are close to each other while the $\overline{g}$ values have wider range.

For the bi-objective optimization problem, there can be a set of non-dominated solutions. A solution is non-dominated (Pareto-optimal) if there is no other solution which can improve at least one of the objectives without worsening any of the other objectives. Considering $\overline{g}$ and $\overline{h}$ of the top 100 non-duplicate plans, we find a set of non-dominated solutions as shown in Table III. We can see 7 non-dominated solutions which can be the best plans for the operation. The decision maker can select plans according to his/her preferences and different circumstances. For example in case of the first plan, the distance (from the initial state to the current state) $\overline{g} = 2708.35$ is a bit higher than the other 6 plans. On the other hand, the sequence of activities within this plan brings more closer to the goal state ($\overline{h} = 814.95$). If we consider the last plan the total distance from the initial state to the current state is minimum ($\overline{g} = 2261.33$) but it is a bit away from the goal state as compared to the first plan. It is a kind of tradeoff between overall consequences of performed actions and goal state. If we look at the results, the $\overline{h}$ values of 7 non-dominated solutions are close to each other but the $\overline{g}$ values have wider range. The set of alternative plans with different distances from the initial state to the current state can be useful in different circumstances to reach the desired end state. The Pareto front represented by NSGA-II is shown in Fig. 8.

*E. Comparison of NSGA-II and A**

In order to discuss the efficiency and effectiveness of NSGA-II, we compare the obtained results with the results reported in [5], that were obtained using A*. Out of 10 000 plans simulated using A*, we select the top 100 based on f values. In case of NSGA-II, we terminate after 2000 plan evaluations and select the top 100 plans based on f value. To compute $\overline{g}$ and $\overline{h}$, we run 20 replications for both. The comparison of the computational results of NSGA-II and A* is shown in Fig. 6 and Fig. 7.

In Fig. 6, we compare $\overline{g}$ of the computational results obtained using these two algorithms. On x-axis we have 100 plans which are sorted based on $\bar{g}$. This Figure shows that NSGA-II is much more efficient than A* in getting good $\bar{g}$ values.

Now we look at Fig. 7 where the comparison is based on $\bar{h}$ (distance to the goal state). On x-axis we have 100 plans which are sorted based on $\bar{h}$. The two algorithms show almost similar results for most of the plans. For the top 24 plans, A* is slightly better than NSGA-II. On the other hand from plan number 74 to 90, NSGA-II is slightly better than A*. Considering the top plans we can say that A* is a little more effective in approaching the goal state.

The Pareto fronts obtained by both algorithms is shown in Fig. 8. The fronts are composed of non-dominated solutions which are obtained using steps mentioned in methodology section III-C. The performance of the algorithms can be also compared using HV values. The higher the HV of obtained solutions the better is the algorithm. The HV of the Pareto-fronts obtained by NSGA-II and A* are 0.57 and 0.50 respectively, which indicates that NSGA-II is better than A*.

Overall we can say that NSGA-II is much more efficient than A* in getting good $\bar{g}$. On the other hand A* is a little more effective in getting good $\bar{h}$ (approaching the goal state).

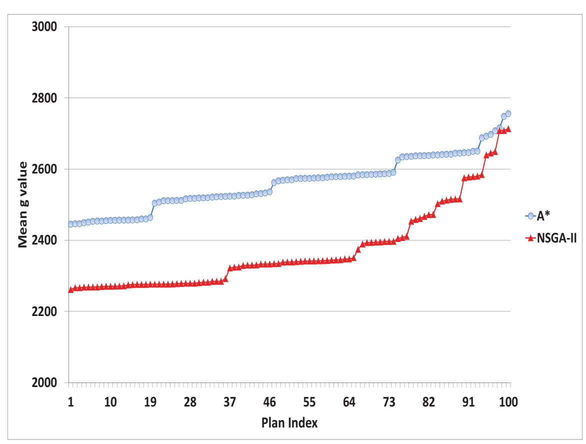

Fig. 6. Mean g-value for top 100 sorted plans based on $\overline{g}$ for both algorithms

TABLE III
PARETO-OPTIMAL SOLUTIONS IDENTIFIED BY NSGA-II

#	Pareto-optimal Plan	Objective Values	Function
		$\bar{g}$	$\bar{h}$
1	{1, 2, 41, 42, 44, 108, 20, 10, 23, 14, 109, 4, 64, 50, 51, 52, 16, 63, 54, 55, 56, 68, 75, 67, 86, 90, 79, 80, 94, 98, 95, 96, 102, 105, 31, 32, 33, 34, 57, 46, 47}	2708.35	814.95
2	{1, 2, 41, 42, 43, 6, 20, 5, 13, 15, 65, 49, 50, 60, 52, 16, 3, 53, 55, 56, 66, 67, 81, 93, 79, 80, 97, 95, 96, 104, 106, 31, 32, 33, 34, 57, 46, 47}	2509.7	815.42
3	{1, 2, 41, 42, 44, 6, 20, 7, 12, 62, 64, 50, 60, 52, 18, 28, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 106, 29, 30, 58, 46, 47}	2338.23	815.785
4	{1, 2, 41, 42, 44, 108, 20, 7, 12, 65, 49, 50, 51, 52, 16, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 104, 107, 29, 30, 57, 46, 47}	2276.47	815.81
5	{1, 2, 41, 42, 44, 108, 20, 7, 11, 62, 49, 50, 51, 52, 17, 63, 53, 55, 56, 66, 67, 89, 79, 80, 97, 95, 96, 101, 105, 29, 30, 57, 46, 47}	2274.535	816.375
6	{1, 2, 41, 61, 43, 6, 20, 7, 12, 65, 64, 50, 60, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 103, 106, 29, 30, 58, 46, 47}	2270.86	816.405
7	{1, 2, 41, 42, 44, 6, 20, 7, 12, 62, 64, 50, 60, 52, 17, 3, 48, 55, 56, 66, 67, 88, 79, 80, 97, 95, 96, 102, 106, 29, 30, 58, 46, 47}	2261.33	816.705

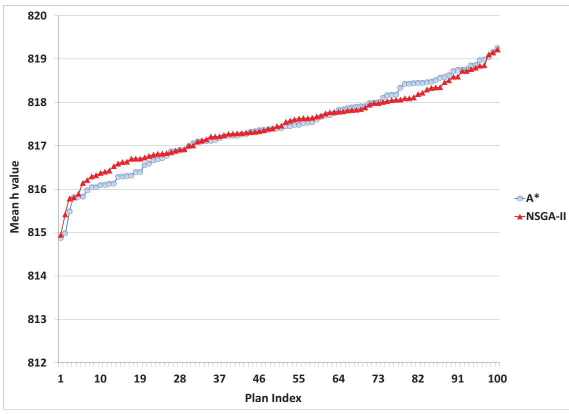

Fig. 7. Mean h-value for top 100 sorted plans based on $\bar{h}$ for both algorithms

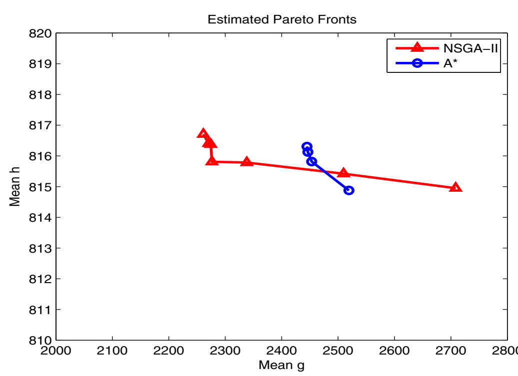

Fig. 8. Pareto fronts

IV. CONCLUSION

In this paper, we formulate the effects-based planning problem as a bi-objective optimization problem and solve it using a widely used multi-objective evolutionary algorithm, NSGA-II.

Considering an expeditionary operation scenario, we simulate a subset of possible plans and present the decision maker with 100 such alternative plans which are capable of achieving a desired end state efficiently. Considering $\bar{g}$ and $\bar{h}$ of top 100 non-duplicate plans, we find a set of non-dominated solutions as shown in Table III. The decision maker can select plans according to his/her preferences and different circumstances.

In order to discuss the efficiency and effectiveness of the algorithm, we compare the results of NSGA-II with the results of A*. In Fig. 6 and Fig. 7, we compare $\bar{g}$ and $\bar{h}$ of the computational results obtained using these two algorithms. Looking at the results, we can say that NSGA-II is much more efficient than A* in getting good $\bar{g}$. On the other hand A* is a little more effective in getting good $\bar{h}$ (approaching the goal state). We use the HV values to compare the two Pareto-fronts shown in Fig. 8. The HV of the Pareto-fronts obtained by NSGA-II and A* are 0.57 and 0.50 respectively, which indicates that NSGA-II is better than A*.

REFERENCES

[1] E. Smith, *Complexity, Networking, and Effects-Based Approaches to Operations*, ser. Future of command and control. CForty-OneSR Cooperative Research Program (C C R P) Publications Series, 2006.

[2] J. P. Hunerwadel, "The effects-based approach to operations: Questions and answers," *Air & Space Power Journal*, vol. 20, pp. 53–62, 2006.

[3] Effects-based Approach to multinational operations, Concept of operations (CONOPS) with implementation procedures, Version 1.0. Suffolk, VA: Unites States Joint Forces Command, 2006.

[4] J. Schubert, F. Moradi, H. Asadi, P. Hörling, and E. Sjöberg, "Simulation-based decision support for effects-based planning," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 636–645.

[5] J. Schubert, F. Moradi, H. Asadi, L. Luotsinen, E. Sjöberg, P. Hörling, A. Linderhed, F. Hinshaw, and D. Oskarsson, "Simulation-based decision support evaluating operational plans - final report," FOI-R–3635–SE, Division of Information and Aeronautical Systems, Swedish Defence Research Agency, Stockholm, Tech. Rep., 2013.

[6] F. Chicano, F. Luna, A. J. Nebro, and E. Alba, "Using multi-objective metaheuristics to solve the software project scheduling problem," in *GECCO*. ACM, 2011.

[7] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, 1999.

[8] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," 1998.

[9] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.

[10] H. Asadi and J. Schubert, "A stochastic discrete event simulator for effects-based planning," 2013, submitted.

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

[12] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011.

[13] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Tech. Rep., 2006.