Multi-objective Genetic Algorithms for Courses of Action Planning

Lamia Belfares
Department of Operations and Decision Systems
Faculté des sciences de l'administration
Université Laval
Québec (Qc) G1R 3X6, Canada
lamia.belfares@fsa.ulaval.ca

Adel Guitouni
Decision Support Systems Section
Defence R&D Canada – Valcartier
2459 Pie-XI North, Val-Bélair, G3J 1X5, Canada
adel.guitouni@drdc-rddc.gc.ca

Abstract - Planning military courses of action is a very complex and difficult activity. Planners should take into consideration environmental information, predictions, the end state targeted and resource constraints. Development of courses of action involves solving simultaneously planning and scheduling problems. In this work, a new approach based on genetic algorithms (GA) and multi-objective optimisation is proposed to support resource-constrained courses of action development where both cardinal and ordinal objectives are considered. A vector of fitness evaluations is proposed to control the proportion of the infeasible solutions. Crossover and mutation operators are designed to diversify the search space and improve solutions on all objectives from one generation to another. In the replacement strategy, a selection procedure, based on the dominance concept and a multi-criteria filtering method, is proposed. Such a strategy is applied when the population reaches a critical size. Different GA schemes are compared and their strengths and weaknesses are discussed. The multi-criteria filtering procedure used in the replacement strategy proved very efficient in the diversification of the Pareto front.

1 Introduction

The Course of Action (COA) Development step of the Military Operational Planning Process (OPP) involves the entire staff. The Commander's guidance and intent helps the staff to focus on the development of comprehensive and flexible plans within the time available. These COAs "should answer the fundamental questions of when, who, what, where, why and how" [CFC Toronto, 2000, US Army, 1997]. Each COA should be suitable, feasible, acceptable, exclusive and complete. A good COA positions the force for the future operations and provides flexibility to meet unforeseen events during its execution. The "who" in a COA does not specify individual units, but rather uses generic assets and capabilities. During the development step, the staff analyses the relative combat power of friendly and enemy forces, and generates comprehensive COAs.

During the mission analysis, the staff should identify the assigned and implied tasks to perform the mission. These tasks can be decomposed into sub-tasks. Tasks and sub-tasks can be represented by means of a hierarchical structure called work break-down structure. Leafs of this hierarchical structure are called elementary tasks. Synchronization analyses lead to identify temporal and spatial relationships between elementary tasks (*e.g.* End-Start, Start-Start, End-End, Time Laps, Same Spatial Zone…). Staff should then consider all available resources and capabilities and assign them to the tasks. Synchronizing COA requires scheduling starting and ending times of all tasks according to resource availability, deployment constraints and task relationships. Any resource or capability has an availability calendar, in-use costing, required preparations, required staffing, etc…

In summary, the challenge for the planning managers is to generate complex, spatially and temporally interdependent activities with precedence relationships, subject to resource constraints, and satisfying multiple incommensurable and often conflicting criteria.

Guitouni et al. (2000, 2002) proposed to model a COA planning as a multiple mode resource-constrained project-scheduling problem (MRCPS) since, from a methodological point of view, planning and scheduling are not much different. The model consists of representing generic activities (tasks with specific combinations of resources) into elementary (or primitive) actions interrelated to accomplish the mission objectives. This process implies the identification of the tasks (when and where) as well as their precedence relationships, the pool of available resources with their localization, and finally the objectives of the mission. A COA is then represented as an oriented time-space graph (see Fig. 1). Depending on the combination of resources allocated and the action's position in the schedule, different COA networks could be obtained. They constitute variants (or alternatives) of a mission with different evaluations on objectives.

Solving COA planning problems is NP-Hard. To obtain promising feasible alternatives with respect to multiple objectives, we thought to explore the potential of evolutionary algorithms (EA), a meta-heuristic that has proven reliable for solving combinatorial NP-hard problems.

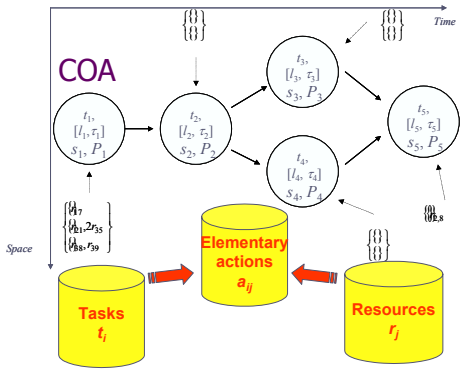

Figure 1: COA graph representation.

Several approaches have been proposed to deal with the multi-objective aspect of resource-constrained project-scheduling (RCPS) problems, and the most common one combines all objectives into one single scalar value by using weighted aggregating functions according to the preferences set by the decision makers. The search is then performed several times to find a compromise solution that reflects these preferences. Another approach is to generate the set of compromise solutions in a single execution of the algorithm such as done by EA. This has increased the interest for the Pareto optimisation and multi-objective EAs. However, applications to scheduling problems are still scarce (Landa Silva and Burke, 2002).

In this work, different COA variants are generated using a specific evolutionary algorithm: the genetic algorithms (GA). GA are able to deal simultaneously with multiple solutions for solving multi-objective optimisation problems. This allows a set of potential Pareto optimal solutions to be found in the same iteration without any sensitivity on the shape or the continuity of the Pareto front (Coello Coello, 1999).

The set of potential compromise solutions (Pareto solutions) is obtained here by investigating different procedures based on dominance concept combined with a new approach referred to as a first order multi-criteria filtering method.

This paper is organised as follows. In section 2, Pareto-optimality is summarised. Section 3 is devoted to the multi-objective GA (mGA) features being developed. Section 4 presents the indicators for performance assessment used to evaluate our algorithm. In section 5, computational results of the proposed approach and a comparison with a method from the literature are analysed leading to some conclusions in section 6.

2 Pareto-based multi-objective genetic algorithms

2.1 Pareto optimality

Research on multi-objective optimisation goes back to the XIX century with Vilfredo Pareto (1896) who introduced the concept of *Pareto optimum*. A vector solution x* is called *Pareto optimal* if and only if x* is as good as all the other feasible solutions with respect to all objectives, and there exists at least one objective for which x* is strictly better than all other feasible solutions. *Pareto* optimum is generally not a single solution but a set of solutions called *efficient* or *non-dominated* solutions. The set of these efficient solutions in the decision space is denoted as the *Pareto set* and in the objective space as the *Pareto front.*

Generally, in combinatorial optimisation, weak dominance is applied because it permits more solutions to be reached from the existing ones with regard to the connectedness of the search space. Moreover, it is impossible to characterise a priori the set of efficient solutions in NP-hard problems. Therefore, we will use the potential or approximation efficient set to represent the Pareto optimal solutions. In the "search and then decision making" approach used here, the goal is to find the largest number of efficient solutions to allow the decision makers (DM) to choose the most appropriate.

2.2 Multi-objective genetic algorithms

Several approaches are used in mGA to improve the generation of solutions. Some of them are non-Pareto approaches such as the vector evaluated genetic algorithm (VEGA) first proposed by Schaffer (1984, 1985), and the lexicographic GA of Fourmen (1985). Others, based on the concept of non-dominance, use the Pareto optimality in their search procedures (*e.g.* Fonseca and Flemming, 1993; Srinivas and Deb, 1995). The Pareto approaches have two goals: the convergence towards the Pareto frontier and the diversification of the solutions spread all over the Pareto frontier. The selection step in the mGA represents the key of success for convergence. Several ranking methods were proposed to rank solutions according to their suitability in order to determine their chance of selection for procreation. The pioneering algorithms are the non-dominated sorting genetic algorithm (NSGA) proposed by Srinivas and Deb (1995) based on the original idea of Goldberg (1989) and the multi-objective optimisation genetic algorithm (MOGA) developed by Fonseca and Fleming (1993). The selection is done using different operations such as the niched Pareto genetic algorithm (NPGA) developed by Horn et al. (1994) based on Pareto domination tournaments, and the elite preserve strategy proposed by Tamaki et al. (1996). Elitism is a particular form of selection used to insure that the best solutions are preserved in the next generation (De Jong, 1975; Goldberg, 1989). In their strength Pareto EA (SPEA), Zitzler et al. (1999) use an external population to store non-dominated solutions that also participate in the selection procedure for reproduction. The diversification can be achieved by artificial niche formation and speciation found by Deb and Goldberg (1989) to avoid "genetic drift". As pointed out by Bosman and Thierens (2003) there is a trade-off to make between a diversified approximation set (diversity) and an approximation set close to the Pareto front (proximity) The different strategies used by the current

state-of-the-art evolutionary methods such as NSGA-II (Deb, 2001), NPGA2 (Erickson et al., 2001) and SPEA2 (Zitzler et al., 2001), based on dominance, elitism and niching or crowding, aim to achieve this two-sided goal. The approach proposed in this work includes these two aspects (convergence towards Pareto frontier and diversification of efficient solutions) by using a first order multi-criteria filter (MFP) to select solutions making the next generation.

3 Construction of a GA-based multiple objectives RCPS

Multi-objective COAs could be characterised by a set of tasks, a set of resources, precedence relationships, resources availability constraints and global performance functions (criteria) *F*.

The problem formulation states as follows:

$$\text{Optimize } F_z,\ z = 1,\ldots Z \quad (1)$$

$$\text{s.t. } t \in D \quad (2)$$

$$\text{s.t. } R \in C \quad (3)$$

with the vector of tasks $t = \{ t_1, t_2, \ldots t_n\}$ having the following attributes for each task t_i :

- Starting and ending time [td(i),tf(i)] considered as integer variables. The earliest and latest starting and ending time, respectively. $[\tau_s(i), \tau_e(i)]$ are used to generate different initial solutions.
- A localization spatial coordinate (x,y,z)
- Type and quantity of resources required, represented by a set **R** composed of renewable and non-renewable resources available in limited quantities. $R_k(t_i) = \{r_{1i}, r_{2i}, \ldots r_{mi}\}$ is the kth set (or combination) of resources required to accomplish the task t_i .
- Set of predecessors {PR} characterized by the tasks that temporally and/or spatially precede t_i.

Resources with the following attributes:

- Starting and ending time of availability $[t_{rs}(k), t_{re}(k)]$ (resource's timetable)
- Quantity available during this interval of time
- Localization of the resources (depot) (x, y, z)
- Type of resource
- Other specific characteristics such as in-use cost, mean speed (for mobile resources), reliability, etc.

Equation (2) concerns the constraints that ensure that each task is processed once in its time interval and precedence conditions are fulfilled (feasible tasks). Equation (3) express the resource constraints (*e.g.* availability).

The mGA is used to find different task-resource combination networks where all activities are completed, the resources and precedence constraints satisfied, and the best compromise between criteria reached. The optimisation is carried out using a randomly initialised population for a low size problem (COAs with less than 10 actions). For large size problems (COAs with more than 10 actions), a heuristic method based on the network approach and CPlex is used (Urli et al., 2003). Crossover and mutation are used for the exploration and exploitation of the search space. The principle of the variable neighbourhood search (VNS) method is used during the optimisation. This is achieved by introducing three binary (crossover, mutation) and one unary (mutation) operators in order to explore a great number of neighbourhoods. Unlike classical GA procedures, the population size is not kept constant from generation to generation but increases by reproduction and mutation until it reaches a critical value determined empirically as explained in section 5.1. A replacement strategy is then applied to select the best candidates for the next generation (survivals). The selection is achieved based on dominance and/or a first order multi-criteria filtering procedure (MFP) detailed in section 3.4 and summarised in Figure 2.

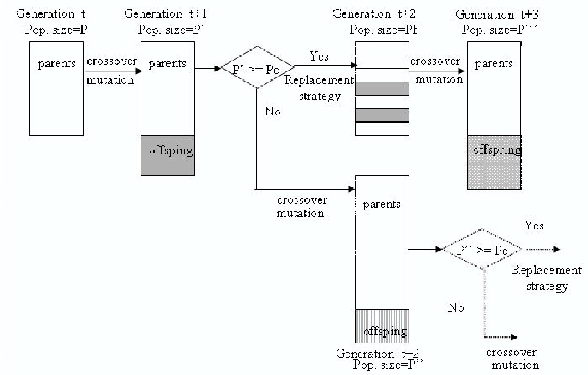

Figure 2. Evolution of the population from generation to generation

3.1 Encoding variables

Each solution, representing a COA network, is encoded in the following form: $x(i) = [(t_1, R_{1,1}), (t_2, R_{2,4})..., (t_n, R_{n,l})]$, for i =1,….popsize, where popsize is the population size. t_j is the jth task to be scheduled and $R_{j,k} = R_k(t_j)$ is one of the sets of resources available to accomplish this task.

3.2 Evaluation of a solution

As it is difficult to apply genetic algorithms to a multi-objective constrained problem, one of the options proposed in the literature is to integrate the constraints into the objective functions using penalty functions. However, the aggregation of penalty functions to the objectives in a highly constrained problem may lead to a large proportion of unfeasible solutions and there is no control of this number. In this work, the constraints are considered as functions to be optimised as the objectives and the solutions generated will be retained regarding both their objectives and constraint violation. Thus, to each solution is assigned a fitness vector.

3.2.1 Constraint functions

Resource availability

Each resource, r_{kj}, used by task t_j must be available, in the desired quantity, during the period $[t_{rs}(j), t_{re}(j)]$.

A penalty coefficient DP(j) associated to each task is introduced and defined by:

$$DP(j) = \begin{cases} 1, \text{if all the resources to accomplish the task } t_j \text{ are available} \\ 0, \text{otherwise} \end{cases}$$

For each solution, it follows:

$$fitn_1(i) = \sum_{j=1}^{n} PP(j) \quad (4)$$

This function is to be maximised (arg [max $fitn_1(i)$] = n) to force the search into in the feasible region.

Precedence

Some tasks have successors and predecessors. Other tasks are free. For each task j, if all its predecessors have been correctly scheduled, we set tPP(j) =1 otherwise tPP(j) = 0. The precedence penalty is defined as:

$$fitn_2(i) = PP(i) = \sum_{j=1}^{n} tPP(j) \quad (4a)$$

As with $fitn_1(i)$, $fitn_2(i)$ is to be maximised.

3.2.2 Objective functions

The objectives considered here could be cardinal (quantitative) such as cost, reliability, make-span, or ordinal (qualitative) such as the impact of a COA. For a uniform optimisation (maximisation), the objectives to be minimised are reformulated and denoted $fitn_j(i)$,j =3,.., Z.

3.3 Genetic operators

Crossover(or recombination)

Selection of two parents for the crossover is done using the roulette wheel selection. Two candidates are selected using the following procedure repeated popsize times:

- Choose randomly two fitness components $fitn_k$ and $fitn_l$ (objectives or constraints)
- Evaluate selection probabilities of the population according to these two fitness components:

$$ps_k(i) = \frac{fitn_k(i)}{\sum_{i=1}^{popsize} fitn_k(i)}, \quad ps_l(i) = \frac{fitn_l(i)}{\sum_{i=1}^{popsize} fitn_l(i)} \quad (5)$$

- Select randomly one candidate per fitness component according to its selection probability.

This selection procedure, favouring one criterion per candidate, aims to produce offspring characterised by the best features present in their parents, and ultimately induces higher population diversity.

Two crossover procedures are proposed, to be used alternatively, in order to explore a greater number of search spaces. The first one is the uniform crossover operator (Syswerda, 1989) which has been shown to be superior to traditional crossover strategies for combinatorial problems. When two chromosomes are selected for crossover, a random mask is generated and their genes are exchanged according to the mask. This mask is simply a binary string with the same length as a COA vector (Sec. 3.1). The parity of each bit determines which genes will be exchanged. The gene in this procedure is represented by an action, $(t_j,R_{j,k})$.

The second operator is the partial mapped crossover (PMX) proposed by Goldberg and Lingle (1985) and is an extension of two-point crossover to permutation representation.

A repairing procedure is used, in both operators, to resolve illegitimacy of the offspring if some activities are missing or duplicated. This procedure is achieved by simply transferring these activities from one child to the other one.

Mutation

Mutation is applied randomly on the population and the probability of mutation is inversely proportional to the population size as recommended by De Jong (1975). The two operators used for the mutation consist of:

- Exchanging, with a probability p_α, a randomly selected combination of resources between two COAs. The offspring, which received the combination with the best criterion, is retained.
- Switching the quantity of two resources in a combination of resources associated to a randomly selected task j. This option is used every four generations.

The probability of mutation p_α of a set of resources $R_{i,j}$ in a chromosome is based on a criterion related to the resource characteristics such as cost or reliability. In this way, only the best resource combination is exchanged, concerning the selected criteria. These probabilities are used alternatively from one generation to the other to produce COAs with improved combinations of resources.

3.4 Replacement strategy

The solutions generated from the crossover and the mutation operations are evaluated. When the population size attains or exceeds a critical value P_c, P_f individuals are selected, based on their fitness vector, among the parents and the offspring using a replacement procedure. Otherwise, new offspring are generated (see Fig. 2). Three strategies are tested for the replacement:

- The multi-criteria filtering procedure (MFP), proposed by Guitouni et al. (2001), returns (a user-defined) P_f non-dominated diversified individuals. MFP is based on multi-criteria dynamic conjunctive and disjunctive procedures. The retained solutions are characterised by at least one best-scored objective or by all objectives achieving minimal threshold values,
- the non-dominated sorting approach (NDS), as proposed by Srinivas and Deb (1995), but used only to rank the best P_f individuals,
- the mixed approach: NDS+MFP procedure.

3.4.1 Strategy 1: MFP method

The filtering procedure is performed in two steps using a disjunctive procedure then a conjunctive procedure. Let A be the set of parents and offspring to be filtered and Â the set of individuals retained for the replacement. Let Card(A) = P_c and Card(Â) = P_f. Let $\underline{e}_0$ be a threshold vector representing the lower limits (lower thresholds) of the objectives imposed to the selection: $\underline{e}_0 = (e_{01}, e_{02}\ldots, e_{0Z})$. Let $\underline{e}_1$ be a threshold vector representing the upper limits (higher thresholds) of the objectives imposed to the selection: $\underline{e}_1=(e_{01}, e_{02},\ldots, e_{0Z})$. S_1: $\{fitn_{1min}, fitn_{2min}\ldots,$

$fitn_{Zmin}\}$ the set of anti-ideal points of A, $fitn_{z,min} = \min_{i=1,Pc} \{fitn_z(i)\}$, z = 1… Z, S_2:{ $fitn_{1max}$, $fitn_{2max}$,…., $fitn_{Zmax}\}$ the set of ideal points of A, $fitn_{z,max} = \max_{i=1,Pc} \{fitn_z(i)\}$, z=1… Z.

Disjunctive procedure

This procedure selects individuals characterised by at least one objective having a maximal value. Let $\hat{A}_d$ be the set of individuals selected by the disjunctive method. The following steps could describe it:

- $\hat{A}d = \emptyset$
- Compute e_{1z} for each objective ($e_{1z} = S_{2z} = fitn_{z,max}$, z =1,…Z)
- Select individuals i that $\exists$ z$\in$ {1,…,Z}, $fitn_z(i) \geq e_{1z}$, i = 1…P_c
- Add these individuals in $\hat{A}_d$.

Conjunctive procedure

Candidates characterised by objective values less than the thresholds are discarded. The thresholds are automatically computed using a dichotomist method between S_1 and S_2. The procedure has the following steps:

- Compute the threshold value e_{0z} for each objective
- Select individual i so that $\forall$ z, $fitn_z(i) \geq e_{0z}$, i = 1…P_c
- Add this individual in the set $\hat{A}_d$ until Card($\hat{A}_d$) = Card($\hat{A}$).

This multi-objective filtering procedure allows the selection of a diversified population of individuals characterised by the fittest objective or all objectives higher than a threshold value.

3.4.2 Strategy 2: NDS method

The solutions are ranked on the basis of non-domination. All non-dominated individuals in the current population are placed at the top of a list and assigned a rank of 1. These solutions are removed from the remaining population and the next set of non-dominated solutions is identified and assigned rank 2. The process is repeated until the entire population is ranked. The top P_f individuals in the list are then selected for the next generation.

3.4.3 Strategy 3: NDS+MFP method

This procedure combines the features of the two previous ones. First, the population is ranked using the NDS method. The set identified with rank 1, *i.e.*, the non-dominated solutions, is selected. Let A_1 be this set.

- If card(A_1) = P_f , then these solutions are individuals of the new generation
- If card(A_1) > P_f, use the MFP method to return P_f solutions
- If card(A_1) < P_f, select the set of solutions identified with rank 2. Let A_2 be this set
- If card ($A_1 \cup A_2$) >P_f, use the MFP method to return P_f solutions
- If card ($A_1 \cup A_2$) < P_f, select the set of solutions identified with rank 3, and repeat the process until P_f solutions are selected.

This strategy differs from NDS because it selects the solutions based on their evaluations besides being non-dominated. The efficiency of these three replacement strategies is examined in section 5.

4 Indicators for performance assessment

Different investigations to evaluate the performance of evolutionary algorithms have been proposed, taking into account the optimisers' stochastic characteristics (see for example: da Fonseca et al., 2001) or not (see for example Zitzler et al., 2003) and often assuming that the Pareto front is known. As pointed out by Bosman and Thierens (2003), comparing the performances of multi-objective evolutionary techniques is not an easy task. There are several criteria of finding a good approximation of the Pareto front and most current methods do not outperform each other but work better regarding different performance indicators. In this work, we examine the performance of the proposed mGA strategies using three indicators, assuming that no a priori information is available on the Pareto set, and taking into account the stochastic aspect of the methods. These indicators are the cardinality of the Pareto set **S** (Van Veldhuizen, 1999), $\mathbf{N_s}$, the diversity of solutions in the decision space, $\mathbf{D_s}$, and the spread of solutions on the Pareto front, **COV**.

Cardinality of the Pareto set, N_s

The non-dominated solutions are stored in an external list updated at each iteration. Thus, the approximation Pareto set **S** is represented by this population when the stopping criterion is reached and $N_s = |S|$.

Diversity of the Pareto set, D_s

When designing a GA algorithm, the diversification of the population (solutions) is often the major goal through the exploration and exploitation operators. Ultimately, the diversity of the solution in the Pareto set represents a desirable result as it gives to the decision maker a larger range of options. The diversity of a population used here is inspired from the Shannon theory to evaluate the value of information based on the entropy H. If the entropy of a message ($0 \leq H \leq 1$) is high, then the redundancy of the information will be low and the value of this message will be high.

Here, the diversity of the solutions is based on the number of alternatives of resource combinations that could be allocated for each task i in the population. So the total number of COA variants is given by the number of different resource combinations available for all the tasks.

$$D_s = \frac{1}{n}\sum_{i=1}^{n} H_i \qquad (8)$$

$$\text{and } Hi = \frac{1}{\log(q_i)}\sum_{j=1}^{q_i}\frac{nij}{pop-size}\log\left(\frac{nij}{pop-size}\right) \qquad (9)$$

where D_s is the diversity of the population, H_i the entropy of a task i, n the number of tasks in the COA, q_i the number of resource' combinations per task found in the population, and n_{ij} the number of solutions (COAs) with different combinations of resources for task i. The higher the number of COAs with different resource configurations per task, the higher the diversity of the approximation set.

Diversity of the Pareto front COV

We propose to express the diversification of the solutions spread all over the Pareto front by the extent of coverage,

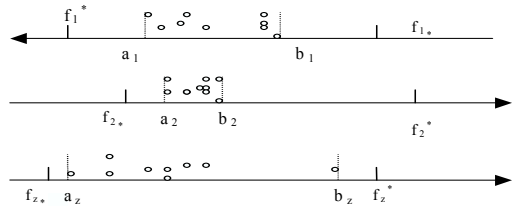

Figure 3. Spread of solutions over the Pareto front in multidimensional space

in each objective's dimension separately as illustrated in Figure 3. This indicates the size of the objective space covered by the Pareto set. The extent of coverage relative to each objective f_i is defined as follows:

$$\mathrm{Cov}_i = \frac{\max_{x,y \in S} \left| f_i(x) - f_i(y) \right|}{\left| f_i^* - f_{i*} \right|} \qquad (10)$$

where the numerator expresses the maximum distance among solutions of the approximated Pareto set **S** in the dimension i, and f_{i*} and f_i^* are, respectively, the anti-ideal and ideal values of f_i (true extremes of the objectives or its worst and best values by comparing simultaneously all the approximation sets).

Zitzler and Thiele (1998) have proposed a similar indicator, but aggregating the Cov_i for i=1,…, Z, by considering the union of all bounding-boxes covered by the Pareto set is misleading. A single, global score for the coverage does not reflect how the diversification is with respect to a given criteria. For a better appreciation of the dispersion, it is more appropriate to consider the coverage array on all the objectives:

$$\mathrm{COV} = (\mathrm{Cov}_1, \mathrm{Cov}_2, \ldots \mathrm{Cov}_Z) \qquad (11)$$

Comparing these coverage vectors could be achieved using dominance analysis, multiple criteria decision analysis or statistical techniques.

5 Planning a course of action: computational results

In order to evaluate the relative performance of the methods, we first check if an approximation Pareto set dominates another set. This set domination is defined as:

S_1 strictly dominates S_2 if $\forall x \in S_2$, $\forall y \in S_1$,

$\forall i, 1 \le i \le Z$, $f_i(y) \ge f_i(x)$ and $\exists$ k, $f_k(y) > f_k(x)$.

We, then compare the difference between their performance indicators and test their significance by using the Wilcoxon signed-rank statistical test since the data distributions are not necessarily symmetrical.

If no conclusion can be drawn, we can examine which method puts more emphasis on diversity or on getting a high number of efficient solutions.

The efficiency of the method proposed here is investigated using the example of courses of action with four objectives: the cost and the make-span to be minimised (cardinal and linear objectives), the resource reliability and the impact on the enemy, to be maximised. The impact is measured on a qualitative scale (ordinal objective). Resource availability and tasks precedence constraints are considered. Three examples are studied to examine the effect of the problem size. The performance indicators of an algorithm are compiled from 5 to 10 runs of each test application. Comparison between algorithms is done based on the means of these indicators. Since the objectives are the cost, the make-span, the reliability, and the impact, the diversity COV of the approximated Pareto set is defined as (Eq. 11): COV = (cov_{cost}, $\mathrm{cov}_{reliabilty}$, cov_{impact}, $\mathrm{cov}_{makespan}$).

In the first example, we consider 6 tasks and 5 generic (types) resources, combined in different sets, to generate randomly 10 initial COAs. This example is denoted by **6t-problem**. The resources are defined by their cost (fixed and in-use cost) and their reliability used to compute the cost and reliability of the COA (Guitouni et al,. 2002). The impact of a COA is calculated as the median value of the actions' impact, which are ordinal values given by the decision makers (Guitouni et al., 2002). In this study, these data are randomly generated. The make-span considered here is the total delay between all the actions in the COA network. In the second example, we consider COAs defined by 50 tasks and 3 generic resources denoted by **50t-problem** and a third example with 100 tasks and 3 generic resources denoted by **100t-problem**. Initial populations of 15 and 22 COAs, for respectively the second and third examples, are created using a heuristic based on the network approach and CPlex (Urli et al., 2003).

For the sake of comparison, we have coded the elitist non-dominated sorting genetic algorithm method (ENGA) proposed by Bagchi (1999) which has been found more efficient to discover the Pareto front compared to the well-known NSGA (Srinivas and Deb, 1995). We have chosen this method because it uses non-dominated sorting as in our approach and the diversification is based on niche formation and sharing fitness based on Pareto ranking. Moreover, in regard to the highly constrained problem treated here, ENGA was found to be easy to implement and less CPU time consuming compared to other efficient algorithms such as SPEA2 or NSGA-II.

The GA algorithm was implemented in C++. The parameters used in the procedure are:

- probability of crossover $p_c = 0.6$,
- probability of mutation $p_m = 1/\mathrm{popsize}$ which gives a better result compared to a fixed value,
- critical population size to apply the replacement procedure P_c,
- population size returned by the filtration procedure P_f,
- stopping criteria gen_{max} = 500 generations (6t-problem), 250 generations (50t-problem), 150 generations (100t-problem).

5.1 Effect of population size on the method's performance

In this section, we examine the effect, on the algorithm's performance, of the parameters P_c and P_f used in the replacement procedure (Sec. 3.4). The values of the

performance indicators N_s, D_s and COV, summarised in Table 1, are calculated for the test application (five runs for each test) where the NDS procedure is used for the 6t-problem.

Table 1. Population size effect on the quality of the final results

$P_c \rightarrow P_f$	Ns	Ds	COV
30→10	1.8	0.73	(7.4; 2.6; 0; 9)
40→20	2.2	0.60	(4.6; 6.6; 5; 10)
60→30	4.7	0.70	(20; 10.5; 7; 22)
P* = 30	3.5	0.73	(13; 11.5; 30; 11.3)

P*: constant size at each generation.

The results show that larger and diversified approximation sets (high N_s, D_s and COV) are obtained when $P_f = 30$. This corresponds to the size of the problem with 6 tasks $\times 5$ combinations of resources. Applying the replacement procedure when the population size reaches 60 (Pc = 60) appears to be more adequate. Similar conclusions are derived with the two other replacement strategies (MFP and NDS+MFP).

5.2 Comparison between methods

First, performance of the three replacement strategies proposed for mGA (mGA-MFP, mGA-NDS, mGA-NDS+MFP) are compared. The results are for $P_c = 60$ and $P_f = 30$. This choice is based on the empirical results presented in Table 1. The mean values of the performance indicators are tabulated on Table 2. Replacement strategy 3 (MFP+NDS) clearly outperforms the MFP and the NDS procedures regarding the number of non-dominated solutions N_s as well as the extent of coverage COV. The multi-criteria-filtering procedure does not generate all non-dominated solutions because some of these solutions are eliminated by the conjunctive procedure. In the NDS replacement method, among the P_f solutions filtered, those, which have the same rank, thus not dominated, are selected randomly with respect to their position in the list. In the MFP+NDS replacement strategy, these solutions are selected in accordance with their objectives by the conjunctive and disjunctive methods even if, globally, they are incomparable (not dominated). Such a strategy leads to a population of better quality, as their evaluations are greater than the threshold values.

Table 2. mGA performance using the three replacement strategies in the 6t-problem

Method	Ns	Ds	COV
mGA- MFP	3.6	0.33	(18.7; 18.4; 0; 19)
mGA- NDS	4.7	0.70	(20; 10.5; 7; 21.7)
mGA- MFP+NDS	6.8	0.79	(26.3; 30; 25; 39)

We have also compared the features of mGA-DNS+MFP to the ENGA algorithm of Bagchi (1999) for the 6t-problem. For ENGA, the population size is fixed at 30 (such as the P_f value in our method) and the niche size, σ_{share}, is set to 5 which gives the best results after several trials between 1 and 5. As stated by the author, the sharing distance is computed in the phenotype space and the coefficient of the sharing function $\alpha = 2$. Five runs are compiled as well for each test application.

mGA- MFP+NDS is also better than ENGA as shown in Figures 4a-c. ENGA is not able to maintain a high number of feasible non-dominated solutions, from one generation to the other, for such a highly constrained problem. Only one run in five generated an interesting solutions set. The constraints in the fitness vector are not always fulfilled: they do not reach their maximum value (Eqs. 4, 4a) even if the solutions are non-dominated having the best-scored dummy fitness. In our method, feasible non-dominated solutions are archived and updated at each generation. This prevents the loss of interesting solutions during GA processing.

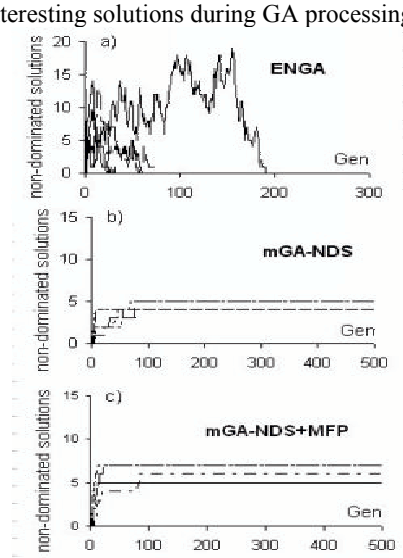

Figure 4. Progress in finding non-dominated solutions on the 6t-problem. 5 runs are compiled for each method.

5.3 Effect of problem size

The efficiency of these methods is also examined large size problems, which are more constrained. For the 50t-problem, $P_c = 150$ and $P_f = 50$ were found to represent the best compromise considering the quality of results and CPU time. For the 100t-problem, $P_c = 150$ and $P_f = 50$ yield the best compromise in terms of CPU time even if higher P_f values generate higher N_S. ENGA, as for the 6t-problem, failed to maintain the non-dominated solutions up to the end of the GA processing. To compare the sharing fitness approach used in ENGA to the MFP procedure, we have modified the former by archiving and updating the approximation Pareto set S discovered at each generation. Mean values for the performance indicators compiled from 10 runs are tabulated in Table 3.

The extent of the coverage on the third objective (the impact) is nil ($cov_{impact} = 0$) for these large-size problems.

This result is the consequence that all non-dominated solutions always score the median value for this objective.

The mGA-NDS+MFP generates a larger number of efficient solutions for both problems and in the 50t-problem a better diversity in the decision and the objective spaces. For the 100t-problem, even if the COV is similar for all the methods, the simulations depicted in Figure 5 illustrate how mGA-NDS+MFP outperforms the modified ENGA in its ability to find better extremes. Data correspond to the 10 processed runs results. Its superiority comes from the multi-criteria filter based on the disjunctive and dynamic conjunctive methods.

Table 3. Algorithm comparison on different problem size

	Ns	**Ds**	**COV**
Algorithm	**50t-problem**		
mGA- NDS	13	0.75	(44; 77; 0; 50)
mGA- MFP+NDS	16	0.76	(68; 94; 0; 34)
modified ENGA	10	0.71	(43; 51; 0; 30)
Algorithm	**100t-problem**		
mGA- NDS	16.7	0.62	(20.7; 13; 0; 18)
mGA- MFP+NDS	22.6	0.61	(17; 11.2; 0; 13)
modified ENGA	12.1	0.64	(18; 9; 0; 12)

6 Conclusion

A new approach is proposed to optimise muli-objective, large-size, highly constrained problem such as COAs planning. This approach, based on genetic algorithms, uses non-dominance and a first order multi-criteria filtering method in the replacement procedure. This approach was compared to the classical one, based on niche formation and sharing function. This is done by analysing the approximation Pareto sets generated by the new algorithm, mGA-NDS+MFP, and by the elitist non-sorting dominated algorithm, ENGA (Bagchi, 1999). Regarding the cardinality and the diversity of these approximation sets, it was possible to conclude that mGA-NDS+MFP is more efficient that ENGA. Moreover, mGA-NDS+MFP does not present the drawbacks inherent to the classical approach such as fine-tuning hard parameters (*e.g.* niche size and distance sharing). By embedding the dynamic conjunctive and disjunctive methods in the replacement procedure, mGA-NDS+MFP was shown to be able to find solutions that are more consistently closer to the Pareto front.

It is also important to undeline the limitations of performance metrics (indicators) used in this paper. In fact, we think that future works should propose new metrics to characterise the Pareto frontier. This charcaterisation could include for example, the dispersion, the shape, and the lower and upper bounds of the frontier.

Acknowledgments

The authors thank Dr N. Lo and Mr Z. Malki for their contribution to implement the algorithms as well as Mr. J. Berger for his technical advices and references.

Bibliography

Bagchi, T.P. (1999) *Multiobjective Scheduling by Genetic Algorithms*, Kluwer Academic Publishers.

Bosman, P.A.N. and Thierens D. (2003) "The balance between proximity and diversity in multiobjective evolutionary algorithms" *IEEE Trans. Evol. Comput.*, vol 7(2), pp. 174-188.

Canadian Forces College Toronto (2000) *Canadian Forces Operations Planning Process CD-Rom*.

Coello Coello, C.A. (1999) "A comprehensive survey of evolutionary-based multi-objective optimization techniques" *Knowledge Inform. Syst.*, vol 1(3), pp. 269-308.

da Fonseca, V.G., Fonseca C.M. and Hall A.O. (2001) "Inferential Performance assessment of stochastic optimizers and the attainment function" *1st Int. Conf. Evol. Multi-Criterion Optimization EMO2001*, E. Ziztler, et al. Eds, Berlin, Germany: Spinger-Verlag, pp. 213-225.

De Jong, A. (1975) *An analysis of the behaviour of a class of genetic adaptive systems*, PhD Thesis, University of Michigan, USA.

Deb, K. and Goldberg D.E. (1989) "An investigation of niche and species formation in genetic function of optimization" *3rd Int. Conf. Genetic Algorithms*, Morgan-Kaufmann Publishers, pp. 97-106.

Deb, K. (2001) *Multi-objective Optimization using Evolutionary Algorithms*, Wiley.

Erickson, M., Mayer A. and Horn J. (2001) "The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems" *1st Int. Conf. Evol. Multi-Criterion Opt. EMO2001*, Ziztler et al. Eds, pp. 681-695. Lecture Notes in Comp. Science, No 1993.

Fonseca, C.M. and Fleming P.J. (1995) "Multiobjective genetic algorithms made easy: selection, sharing and mating restrictions" *IEEE Int. Conf. on Genetic Algorithms in Eng. Syst.*, UK, pp. 45-52.

Fourmen, M.P. (1985) "Compaction of symbolic layout using genetic algorithms" *Int. Conf. Genetic Algorithms and their Applications*, J.J. Grefenstette Ed., Pittsburgh, pp. 141-153.

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

Guitouni, A., Urli B. and Martel J.-M. (2002) "Course of action planning: a project based modelling" *Management Sciences*, submitted.

Guitouni, A., Bélanger M. and Berger J. (2000), *Report on Final Lance Exercise : Canadian Forces College*, DREV-TM-2000-21.

Guitouni, A., Belacel N. and Martel J.-M. (2001) " Procédure de filtrage multicritère de type I" In: *Développement de Méthodes d'Aide à la Structuration Décisionnelles Opérationnelles*, Projet W7701-0-2204.

Horn, J., Nafpliotis N. and Goldberg D.E. (1994) "A niched pareto genetic algorithm for multiobjective optimization" *IEEE World Cong. Comp. Intelligence*,

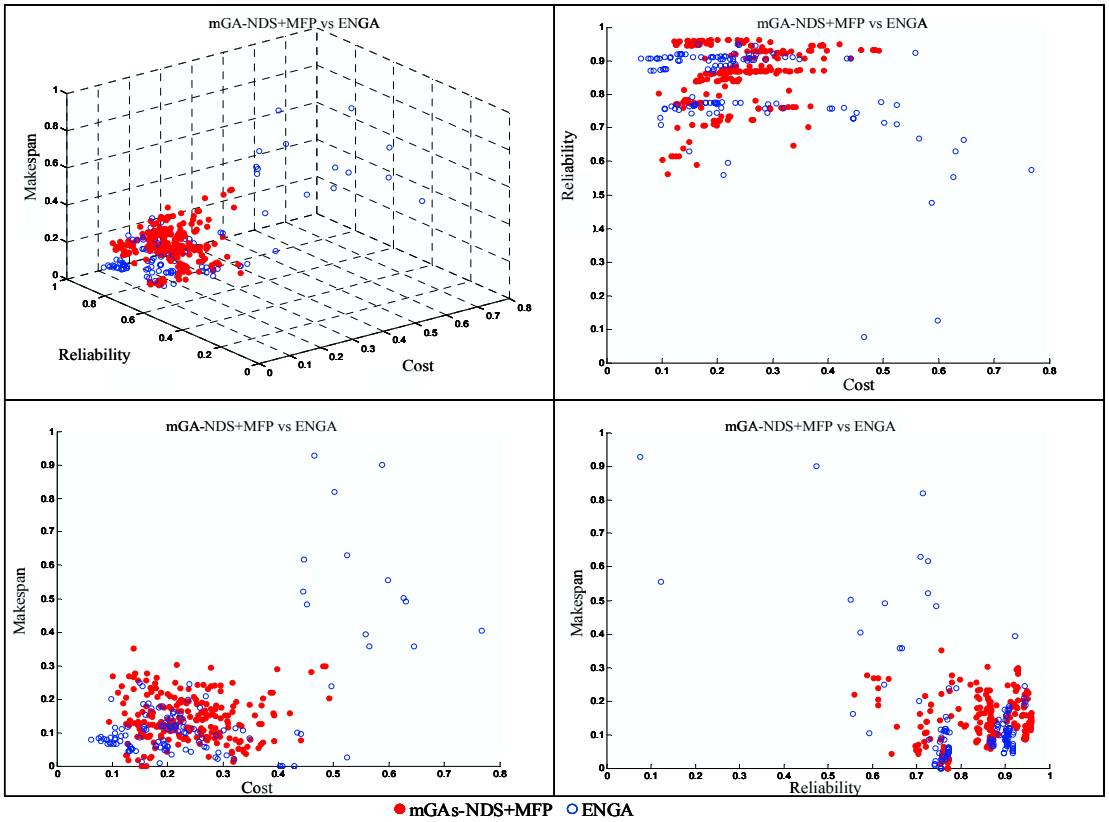

Figure 5: Empirical comparison of mGA-NDS+MFP (filled points) and modified ENGA (empty points) for the 100t-problem

vol 1, pp.82-87.

Landa Silva, J.D. and Burke E.K. (2002) "A tutorial on multiobjective metaheuristics for scheduling and timetabling" 1st *workshop on Multiobjective metaheuristics*, Paris France, November.

Pareto, V. (1896) *Cours d'économie Politique*, vol I and II, F. Rouge, Lausanne.

Schaffer, J.D. (1984) *Some experiments in machine learning using vector evaluated genetic algorithm,* PhD Dissertation (TCGA file No. 00314), Vanderbilt University.

Schaffer, J.D. (1985) "Multiple objective optimization with vector evaluated genetic algorithms" *1st Int. Conf. Gen. Algorithms*, J.J. Grefenstette Ed., pp. 93-100.

Srinivas, N. and Deb K. (1995) "Multiobjective optimization using non-dominated sorting in genetic algorithms" *Evol. Comput.*, vol 2(8), pp. 221-248.

Syswerda, G. (1989) "Uniform crossover in genetic algorithms" *3th Int. Conf. Genetic Algorithms*, J. Schaffer Ed., Morgan Kaufmann publishers, Sam Mateo, CA, pp. 2-9.

T'kindt, V. and Billaut J.-C. (2002) *Multicriteria scheduling: Theory, Models and algorithms*, Springer.

Tamaki, H., Kita H. and Kobayashi S. (1996) "Multiobjective optimization by genetic algorithms: A review" *IEEE Int. Conf. Evol. Comput. ICEC'96*, pp. 517-522.

Urli, B., Lo N. and Guitouni A. (2003) "Une approche d'optimisation dans la génération de Plans d'Actions (COA)" *31th Annual ASAC Conference*, Halifax, NovaScotia, Canada, June 14-17.

US Army (1997) Headquarters, Department of the Army. *Staff Organization and Operations*. FM 101-5, Washington DC.

Van Veldhuizen, D.A. (1999) *Multi-objective evolutionary algorithms: Classifications, analyzes, and new innovations*, PhD Thesis, Dep. of Elect. & Eng, Graduate School of Eng., Air Force Institut of Technology, Ohio, USA.

Zitzler, E. and Thiele L. (1998) "An evolutionary algorithm for multiobjective optimisation: the strength pareto approach" *TIK-Report No. 43*, Institute für Technische Informatik and Kommunikatiosnetze, ETH Zürich, 40p.

Zitzler, E. and Thiele L. (1999) "Multiobjective evolutionary algorithms: A comparison case study and the strength Pareto approach" *IEEE Trans. Evol. Comput.*, vol 3(4), pp. 257-271.

Zitzler, E., Laumanns M. and Thiele L. (2001) "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization" *EUROGEN 2001-Evolutionary Methods for Design, Optimization and Control with Applications to Industrial problems*, Barcelona, Spain.

Zitzler, E., Thiele L., Laumanns M., Fonseca C.M. and da Fonseca V.G. (2003) "Performance assessment of multiobjective optimizers: an analysis and review" *IEEE Trans. Evol. Comput.*, vol 7(2), pp.174-188.