

Research Article

Using Genetic Algorithms to Represent Higher-Level Planning in Simulation Models of Conflict

James Moffat and Susan Fellows

PCS Department, The Defence Science and Technology Laboratory (Dstl), Grenville Building, Fareham PO17 6AD, UK

Correspondence should be addressed to James Moffat, jmoffat@dstl.gov.uk

Received 20 April 2010; Accepted 25 July 2010

Academic Editor: Cyrille Bertelle

Copyright © 2010 J. Moffat and S. Fellows. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The focus of warfare has shifted from the Industrial Age to the Information Age, as encapsulated by the term Network Enabled Capability. This emphasises information sharing, command decision-making, and the resultant plans made by commanders on the basis of that information. Planning by a higher level military commander is, in most cases, regarded as such a difficult process to emulate, that it is performed by a real commander during wargaming or during an experimental session based on a Synthetic Environment. Such an approach gives a rich representation of a small number of data points. However, a more complete analysis should allow search across a wider set of alternatives. This requires a closed-form version of such a simulation. In this paper, we discuss an approach to this problem, based on emulating the higher command process using a combination of game theory and genetic algorithms. This process was initially implemented in an exploratory research initiative, described here, and now forms the basis of the development of a “Mission Planner,” potentially applicable to all of our higher level closed-form simulation models.

1. Introduction

Since the Cold War period, the scenario context has widened considerably, reflecting the uncertainties of the future. Moreover, decision cycles for our customer community in the UK Ministry of Defence (MoD) have significantly shortened. The focus of war has also shifted from the Industrial Age of grinding attrition to the Information Age, as encapsulated in the term Network Enabled Capability (NEC). NEC is a key goal for the MoD, with the emphasis on command, the sharing of awareness among commanders, and the creation of agile effects. These influences together have led to the need for simulation models which are focussed on command rather than equipment, which can consider a large number of future contexts, and which can robustly examine a number of “what if” alternatives [1].

In response to these demands, we have built a new generation of simulation models, with command (and commander decision making in particular) at their core [2]. These span the range from the single environment (e.g., a land only conflict at the tactical level) to the whole joint campaign, and across a number of coalition partners [3].

They also encompass both warfighting and peacekeeping operations. These models have been deliberately built as a hierarchy, feeding up from the tactical (or systems) level to the operational (or system of systems) level, to give enhanced analytical insight, as shown in Figure 1.

2. The Wargame Infrastructure and Simulation Environment

As part of these development activities, we have constructed a stochastic wargame called “WISE” (Wargame Infrastructure and Simulation Environment) [4]. As the name suggests, this is more than just a single model and in fact provides a modelling infrastructure from which a number of tailored models can be created. The key development thus far has been the wargame itself [4–6]. However, a logistics simulation has also been developed and is being used to examine vehicle reliability and consequent repair.

The model addresses a previous gap in modelling capability relating to the representation of command decision making and has utilised where possible novel techniques

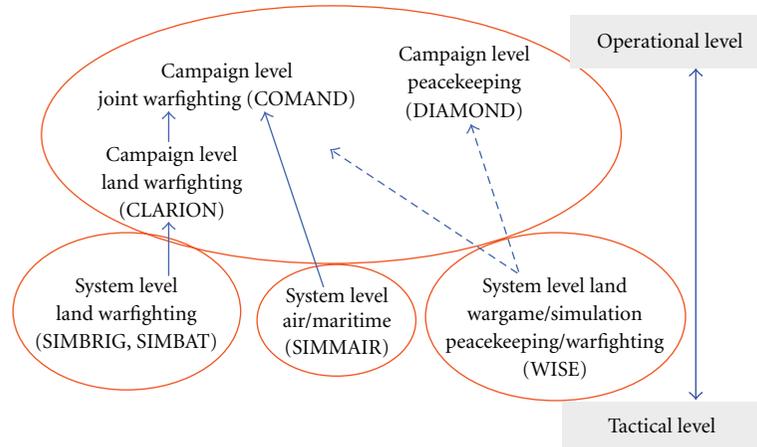


FIGURE 1: The hierarchy of key simulation models, indicating the place of the WISE model in this schema.

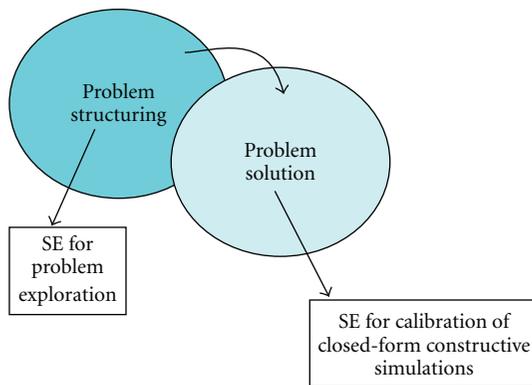


FIGURE 2: Synthetic Environments in problem exploration mode, calibration mode, and transitioning from one mode to the other.

to represent this key aspect of Network Enabled Capability. The wargame represents operations up to Army Division level. Army commanders play the roles of Division and Brigade commanders in the game, on both sides, and they are supported by an underlying simulation environment which represents the evolution of events. This Synthetic Environment (SE) representation exploits the Rapid Planning process [7] to determine the decisions made by the lower-level commanders that are not explicitly represented by players. We define a Synthetic Environment (SE) as consisting of real and simulated people interacting with simulated environments. In contrast, a closed-form constructive simulation consists of simulated people (i.e., computer algorithms) interacting with simulated environments, with no human intervention during the model run. Synthetic Environments are particularly good at exploring new situations and future contexts, as indicated in Figure 2.

In problem exploration, SEs give a rich understanding (i.e., qualitative information) of a small set of possible options. The number of options which can be explored, however, is limited due to the high cost and time required to stage such wargaming events. In order to allow us

to explore around these initial options and thus develop a wider understanding of their robustness (a key aspect of understanding force “agility”), we needed to develop a closed-form discrete event simulation equivalent of the WISE wargame—in essence replacing the human players by some form of artificial intelligence representation, to allow the running of the scenario without human intervention. This was done by exploiting the Deliberate Planning Process [7], an algorithmic representation of higher level command based on a combination of game theory and genetic algorithms. Our implementation was exploratory, and a test of the feasibility of generating ‘sensible’ higher level plans, within a realistic conflict context, using genetic algorithms rather than expert military players. Using the same model as both an SE and a closed-form constructive simulation has the additional benefit that the algorithms derived for planning in the closed-form model can be calibrated by running experiments using expert military players in the SE version of the same situation, as also indicated in Figure 2.

3. Genetic Algorithms

Genetic algorithms are one of a class of evolutionary optimization techniques. These use search strategies across the space of feasible solutions which emulate Darwin’s theory of evolution and natural selection. Genetic algorithms are robust and can be applied to a wide range of optimization problems, as illustrated in [8–14]. This is due to the generic nature of the algorithms used. The process starts by developing a fitness function for each potential solution. This is a transformation which, for each solution, assigns (by some means) a numerical value representing the goodness of that solution. For example, in our case, a high fitness would represent a plan which delivered a good outcome for the commander in terms of casualties avoided or in terms of getting to his objective in a more timely way. Since each solution has a fitness associated with it, the whole space of solutions is sometimes called a fitness landscape. Our aim, in this application, is to traverse this landscape in an efficient

way in order to get to a solution which is sufficiently good (although not necessarily optimal).

In the general approach, and in our application, each solution is encoded as a *chromosome*. This is a binary string of 0 and 1 which encapsulates the numerical characteristics of that solution (e.g., in our case, the number of units deployed and their area of deployment). We employ *haploid* chromosomes, which means that each solution is represented by a single binary string. In some applications, such as the classic work of Hillis on sorting processes [15], *diploid* chromosomes, consisting of pairs of binary strings are used, if the problem characteristics require such a representation. An initial set of feasible solutions (the initial *gene pool*) is generated (in our case 100 random plans are created to form the initial gene pool). Solutions (*parents*) are then drawn from this gene pool in pairs and evolved to produce *children*. Usually, two parents will produce two children, which then, in our case, replace their parents in the next generation of the gene pool. We found this, through testing, to be a sensible way of evolving the gene pool. Alternative replacement options include having the children replace two parents who have a lower fitness value. The selection of parents from the gene pool, and their evolution to produce the child solutions, can also be performed in a number of ways. In our application, we have used *roulette-wheel sampling* for the parents. Thus parents are selected with a probability which is proportional to their fitness value. Again, through testing and the use of military judgement, this appeared to give sensible results. Other approaches which are used include tournament pairing. In this case, two parents are chosen completely randomly. Comparison of their fitness levels leads us to select one of these parents. A similar competition or *tournament* results in another parent, and these two parents then produce two child solutions. This process can be extended to tournaments involving n parents. We considered these n -tournament approaches to be overly complex for our application.

Evolution of the parents to produce the child solutions involves two processes, known as *crossover* and *mutation*. Crossover mimics the way that real chromosomes evolve in a cell through swapping pieces of their genetic code. With a *single crossover operator*, the two parent chromosomes are lined up together, and an incision point is chosen. The cut ends of the chromosomes are then swapped over. The incision point could be chosen at random. In our application, for simplicity and to avoid destroying a potentially good plan through too much randomization, we employ a single crossover operator with a fixed incision point. Whether the cut is applied or not is determined by a fixed probability (the default value we use is 0.7). It is possible to have more than one incision point, but as already hinted, this can lead to too rapid evolution of the chromosome, and the rapid destruction of potentially valuable “genes”, that is, useful parts of the chromosome. The coordinated evolution of such genes was a key concern of Holland [8, 9]. Having applied the crossover process, we then apply the mutation process to each of the resultant pair of chromosomes. This involves flipping (with a fixed probability) each of the binary values of the chromosome from a 0 to a 1 or vice versa. The default probability we use

is set at 0.033. Set too high, this again leads to too rapid an evolution, destroying potentially valuable plans; set too low, the gene pool tends to stagnate. Finally, in moving from one generation of gene pool to the next, we employ a form of “elitism” to ensure that good plans developed on the way are not lost in later evolution. Here the best plan developed so far (in terms of fitness) is carried forward and is considered together with the best plans from the final generation, before a final choice of plan is made. From all of the above, it is clear that although the genetic algorithm is a generic process, there are a number of choices to be made in order to fit the approach to the precise constraints and context of the problem domain.

4. The Deliberate Planning Process

The Deliberate Planner emulates the “formal estimate process” whereby a high level commander develops an overall plan for the campaign. At this level of the command process, a “Blue” (friendly) commander considers a number of potential courses of action, taking into account his intent (i.e., his primary goal or objective) and the intent of the enemy (“Red”) force. The algorithms which we have implemented to represent this process firstly develop a “Recognized Picture” of the layout and intent of the enemy force, based on sensor inputs and information sharing. More details of how this is structured will be given later. On the basis of this Recognized Picture, the planner then decides on a layout of the friendly force which best achieves the commander’s goals. It does this by “breeding” plans in an innovative way, using the genetic algorithm approach described above and then selecting a plan with a high “fitness” level. This is a “satisficing” approach to finding a good plan, reflecting the uncertainty and bounded rationality which is a part of the problem context.

A plan, in this context, is an allocation of a number of units of force to different potential areas of operation across the whole “theatre” of operations. This is turned into a haploid chromosome by expressing each such allocation to a specific region in binary terms, so that the plan is then a string of binary numbers. The fitness of the plan is calculated using a number of historical analysis equations, exploiting the approach of [16], which relate force layout to potential campaign outcome. More precisely, for each allocation of force units to particular areas of operation, these force levels are combined with perceptions (via sensor systems and information distribution) of the enemy force levels in the same areas of operation. These values are then used to calculate outcome values such as the force density ratio. This is a quantitative measure of the relative force balance of own and enemy forces in a given area of operations. These values are then modified by a number of human factors to reflect the cultural background of the forces, and their resilience to the effects of operational shock and operational surprise. These factors have been derived through extensive regression analysis of historical maneuver-based conflicts which have the same general characteristics as the problem domain. From these values, we can in turn calculate key outcome aspects of the plan solution. These are the likely level of

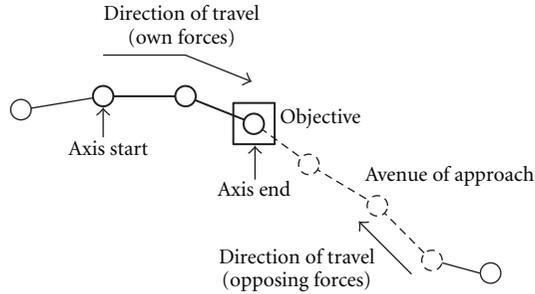


FIGURE 3: Illustration of an axis and objective for a group of “Blue” (friendly) forces with a single objective. Also shown is a possible avenue of approach for a coherent set of “Red” (opposing) forces sharing the same objective.

casualties (for both own and opposing forces), the likely rate of advance towards the designated military objectives (for both own and opposing forces), the probability of immediate breakthrough resulting in a rapid and significant operational gain, and the probability that the plan will lead to overall success in meeting military objectives. Each of these quantitative measures of the “goodness” of the plan contributes separately to the plan fitness function. They are weighted, and these weights can be varied to represent different “styles of command” (e.g., a risk averse commander might put a high priority on keeping casualties to a minimum, while another commander might put more priority on getting to the objective).

The allocation of forces by a plan is defined by a number of axes along which the commander will plan to deploy his forces. The number of axes will be small—3 or 4 is a reasonable number at this level of planning. At the end of an axis is an objective. The objective is the location the plan wishes the allocation of forces to reach on the axis. The objectives defined by each side’s plan need not be the same; they can be asymmetric, due, for example, to different overall plan intents. Each axis also has an avenue of approach. This is also a route leading to the objective of the axis but this time from the perspective of the opposing side; it represents the plan’s perception of a possible way for the opponent to reach the objective on the axis.

4.1. The Recognized Picture. As far as a plan is concerned, its only perception of the outside world is what is in its Recognized Picture (RP). An axis is represented in the RP by a set of networked nodes and links (shown in bold in Figure 3). An axis starts at one RP node and end on another node; all intermediate nodes and links will be part of the axis. The node at which an axis ends is the objective of that axis. Similarly, the avenues of approach to the axis objective are represented by a set of nodes and links in the RP (shown dotted in Figure 3). The avenue of approach starts at one RP node and ends on the axis objective node; all intermediate nodes and links are part of the avenue of approach. The network of nodes and links constituting an axis (including the objective) together with the nodes and

TABLE 1: Game payoff matrix P . This shows the possible courses of action available to the planner, denoted by the rows O_i . The perceived set of courses of action available to the opposing forces is shown as the columns E_j . For each combination of O_i and E_j , we have the payoff value P_{ij} with a positive payoff corresponding to a gain for the planner and a negative value corresponding to a gain for the opposing forces.

	E_1	E_2	\dots	E_S
O_1	P_{11}	P_{12}	\dots	P_{1S}
O_2	P_{21}	P_{22}	\dots	P_{2S}
\vdots	\vdots	\vdots	\vdots	\vdots
O_M	P_{M1}	P_{M2}	\dots	P_{MS}

links constituting the associated avenue of approach form a higher-level structure that we call a “channel.”

There are just two types of operation available to the plan—attack and defend—and a side will do one or the other: the attacker side will attack on each axis defined by its plan, whilst the defender side will defend on each axis defined by its plan.

4.2. Game Theoretic Approach. The core of the Deliberate Planning process model is based on ideas from game theory—the mathematical theory of decision making in conflict situations. Game theory was chosen as the starting point for the Deliberate Planning process model because the theory addresses one of the central elements of the Deliberate Planning process, namely, the analysis of opposing courses of action.

In the following, *planner* refers to the agent in the simulation model that is actually doing the Deliberate Planning. The planner is pitted against an opponent—the *enemy*. Consider the game payoff matrix, denoted by P , shown in Table 1.

Here, the rows (columns) represent different courses of action available to the planner (enemy). O_i denotes the i th course of action available to the planner, and E_j denotes the j th course of action available to the enemy. We define a course of action (CoA) to be a particular (ground and air) force allocation to each objective considered in the planning process. It is important to recognize that in this usage of game theory the E_j represent only the planner’s *perception* of the courses of action that the enemy could follow. Thus, the E_j need not necessarily to reflect what the enemy is actually contemplating doing nor necessarily contain the course of action that the enemy will actually take. The quality of the E_j , in terms of how well they predict future states of the conflict, depends on the ability of the planner to judge the enemy’s physical capabilities and to divine his intentions.

The interactions of the opposing courses of action are represented by the contents of the matrix—the payoffs, P_{ij} . P_{ij} is the payoff from the enemy to the planner that will occur if the planner takes course of action O_i and the enemy takes course of action E_j . A negative value of P_{ij} denotes a payoff from the planner to the enemy. By convention, in the game the planner is attempting to maximise the payoff whilst the

enemy is attempting to minimise the payoff. The payoff can be thought of as a measure of fitness of the planner playing a particular own CoA against a particular enemy CoA. Each planner—one on the attacking side, one on the defending side—will have a different payoff matrix, representing each planner’s *perception* of the possible CoAs open to himself (the O_i) and his opponent (the E_j) and the consequences of interactions between them (the P_{ij}).

A game payoff matrix, of the form shown in Table 1, thus encapsulates measures of fitness for all possible interactions between opposing (own and enemy) courses of action. The essence of the Deliberate Planning model is the analysis, by the planning process, of this payoff matrix (representing the analysis of opposing courses of action) and the selection of a single CoA, O_i , that, in some sense, is the “best” CoA to take, given the perceived enemy capabilities and intentions. The selection of a CoA is the command decision and is a key output of the Deliberate Planning process model, for a given plan.

There are a number of strategies that can be used to analyse the payoff matrix and select the “best” CoA. These strategies represent decision making under certainty, decision making under risk, and decision making under uncertainty. Which strategy is appropriate, in a given situation, depends upon the degree of knowledge the planner has about the enemy’s intentions (the E_j). The strategy most likely to be relevant in a military planning situation, and the one implemented in the Deliberate Planning model, is the last of these, namely, decision making under uncertainty. Within this strategy, there are several different ways of defining the fittest CoA, depending on the criteria used to measure fitness. Four such criteria (termed decision criteria) that can be used are the criterion of pessimism (maximin); the criterion of optimism (maximax); the criterion of least regret; the criterion of rationality. The Deliberate Planning algorithms use the first and second of these criteria—the criterion of pessimism (also known as the maximin criterion or the Wald criterion) and the criterion of optimism.

As an example, consider the strategy of decision making under uncertainty combined with a Wald decision criterion, resulting in a payoff matrix analysis process that represents a conservative decision making approach in which the planner looks for the (own) CoA which offers the *best guaranteed payoff*. This is done as follows. The decision maker determines the guaranteed payoffs by asking, for each CoA: what is the worst that can happen if I use this CoA? For a given CoA, O_i , the guaranteed payoff, denoted by GP_i , is given by

$$GP_i = \min_j (P_{ij}), \quad (1)$$

and the planner then selects the O_i for which the corresponding GP_i is greatest. The decision-maker that chooses this CoA can do so knowing the payoff will be at least

$$\max_i GP_i. \quad (2)$$

no matter what course of action the enemy adopts, the Wald decision criterion minimises the risk involved in making a

decision. It is also referred to as the maximin criterion since the minimum payoff for each of the planner’s own CoA is found first and then the CoA chosen is the one which yields the maximum value of these minimum guaranteed payoffs. This, then, is the core of the Deliberate Planning process model, namely, establish a planner-enemy CoA interaction (payoff) matrix and analyse this matrix using the strategy of decision-making under uncertainty with a maximin or maximax decision criterion. The planning process implemented in the WISE model uses a maximin approach. Figure 4 presents a diagrammatic view of the operation of the planner, taking the place of a higher level military player in the SE version of the model.

5. Testing the Genetic Algorithm in WISE

The Deliberate Planner algorithms, incorporating the genetic algorithm, have been initially implemented as a series of standalone C++ classes in WISE to improve the potential for reuse in other models. A WISE specific interface has been developed to allow it to be run as part of the decision making representation. It is implemented for both sides in the model (i.e., both Blue and Red) although the description here focuses on the Blue side. The interface has been defined such that the data stored within WISE can be packaged generically and passed in to the algorithm in a standard format. Figure 5 shows a high-level process diagram of the implementation within WISE.

In order for a plan to be generated there is an implicit assumption that the unit (the planner) undertaking Deliberate Planning has an understanding (the Recognized Picture) of what is happening around it in the model. This is shown in the top part of Figure 5 labeled “Intelligence Fusion,” where a number of sensor platforms or units are feeding information in to allow the Recognized Picture to be compiled. At the beginning of a run, an “Assess Current Situation” task is called which sends out the initial orders to the sensors to search for information updating the picture. An intelligence fusion process and possible additional tasking of sensor units to add further information are then carried out to further build up the picture and allow an analysis of enemy intent and likely courses of action (i.e., Red strategies in the game theory sense) to be completed. All of the sensor acquisitions are made using the “Surveillance and Target Acquisition” model in WISE which are passed to the planner unit. When a sensor asset completes a search of its tasked zone a “fused” set of acquisitions is passed into the Intelligence Fusion process, and a new order is generated for that sensor asset.

As already discussed, a number of cycles of intelligence fusion are required in order to build up a suitable picture against which to create a plan. Two criteria are specified in the data that determine when intelligence fusion is deemed to be complete enough for planning: (a) the number of times that specified zones must be searched or (b) a time period. The first of these criteria to be realised is used to initiate the plan generation task. We also normally assume that one side in the model is attacking, and the other defending, with the attacker (either Blue or Red) being the first to formulate a plan, followed by the defender.

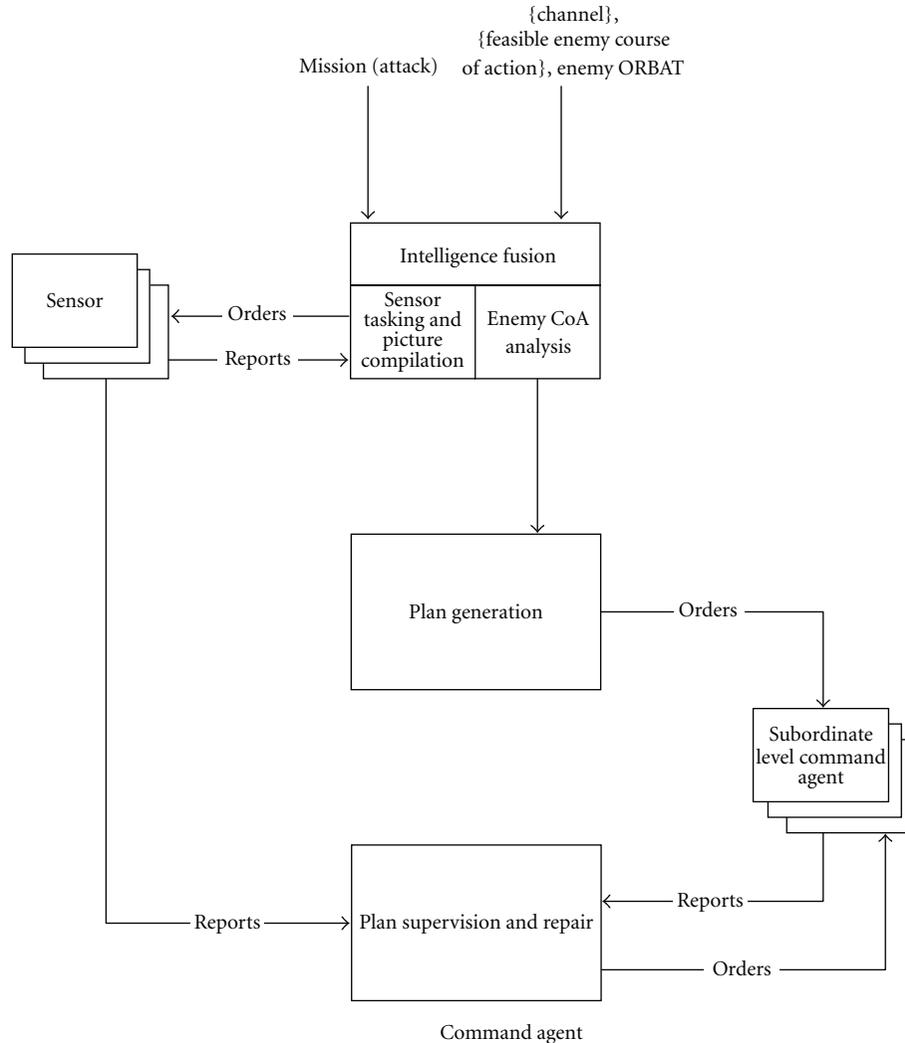


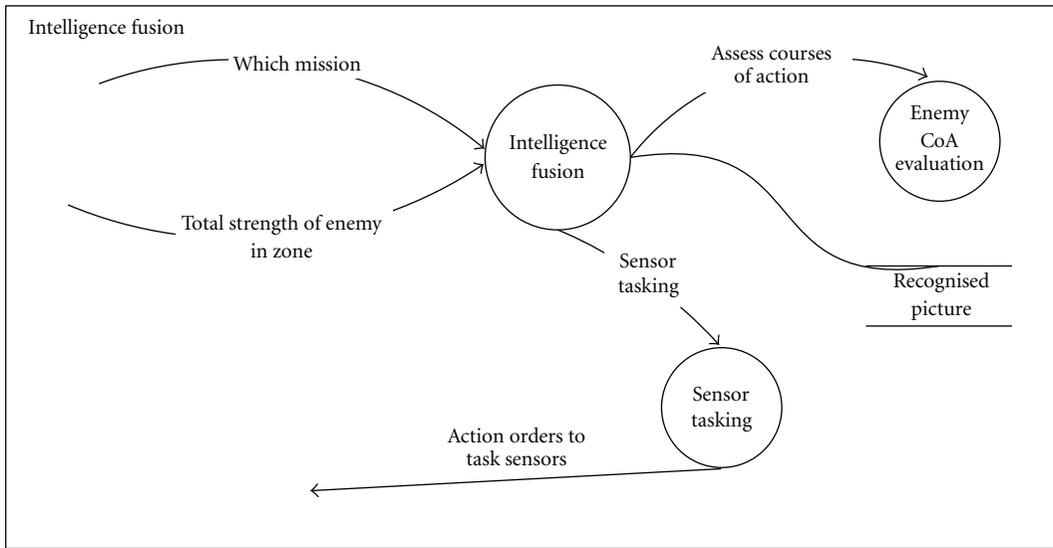
FIGURE 4: Implementation of the Deliberate Planner within a command agent. This illustrates how the various algorithmic modules combine to create a plan.

Once started, the Blue plan generation process takes account of the likely Red strategies and possible own Blue strategies together with the assumed style of command (maximin, for example) in order to determine the course of action to adopt and hence the orders that need to be issued. A plan (a Blue strategy) is a force allocation to a number of areas of operation, and this is evaluated using the plan fitness function, given the possible set of Red strategies. The initial set of Blue plans is then ‘bred’ using the genetic algorithm, as previously described, to determine the best plan to adopt. Once this process is complete, a set of orders are generated and picked up by the interface classes to be translated into the orders required to task units within WISE.

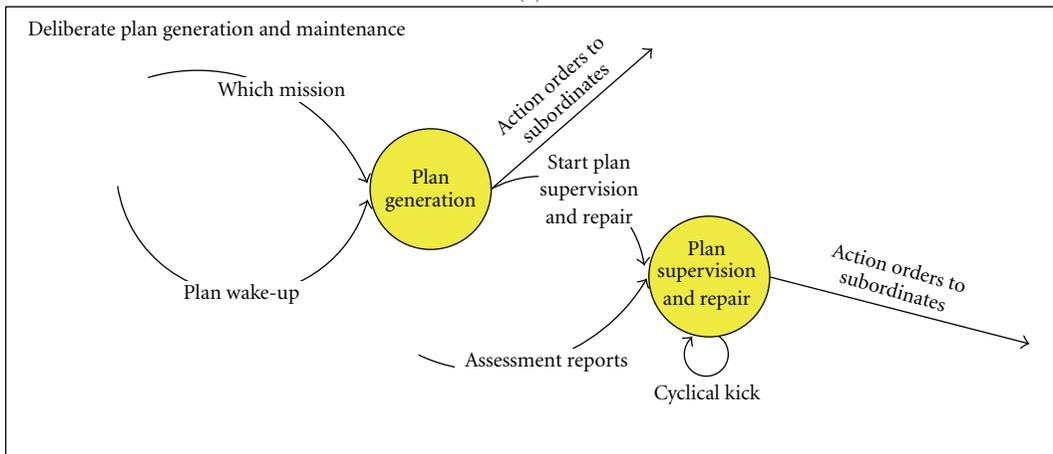
As the plan is executed in the simulation, sensor assets continue to search for further information, and the planner’s Recognized Picture continues to be updated. Each time that this process is carried out, an assessment is made (the ‘Plan Supervision and Repair’ process) to determine whether the plan is performing within defined bounds. This is done by

applying the plan fitness function to the Blue plan as it evolves through the simulation, taking account of additional sensor-based information (i.e., Blue’s evolving perception) about the location of both Blue and Red units. If the plan is failing (i.e., not achieving the required fitness level), the Plan Supervision and Repair process takes place. The planning algorithm determines which areas of operation are failing to meet the plan. It also identifies which units are surplus or in reserve and places these in an availability pool. The areas of operation that are in deficit are then supplemented as required, and a new set of orders are issued.

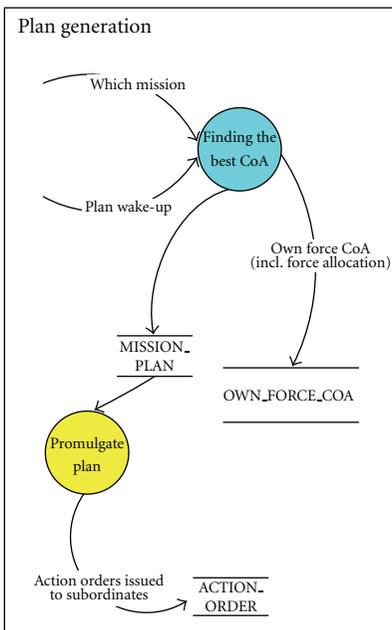
5.1. The Turing Test. In order to test the genetic algorithm, we played through a future scenario using the SE version of WISE, employing expert military players on both the Blue and Red sides. We also represented the same scenario within the closed-form constructive simulation version of WISE and used the military players (who were outwith the project team) to assess the result. We also played through



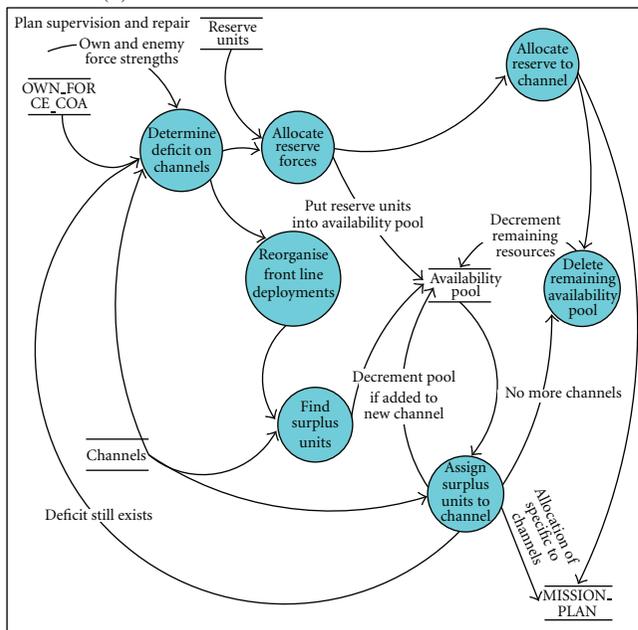
(a)



(b)



(c)



(d)

FIGURE 5: Deliberate Planning and genetic algorithm process model, illustrating how the intelligence fusion, plan generation, and plan repair processes combine together.

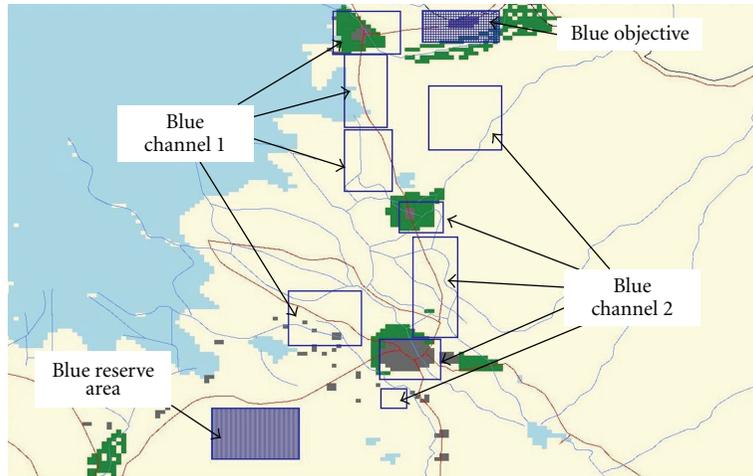


FIGURE 6: Blue areas of operation (channels) are shown, within the context of Blue intent (the Blue Objective to the north of the map).

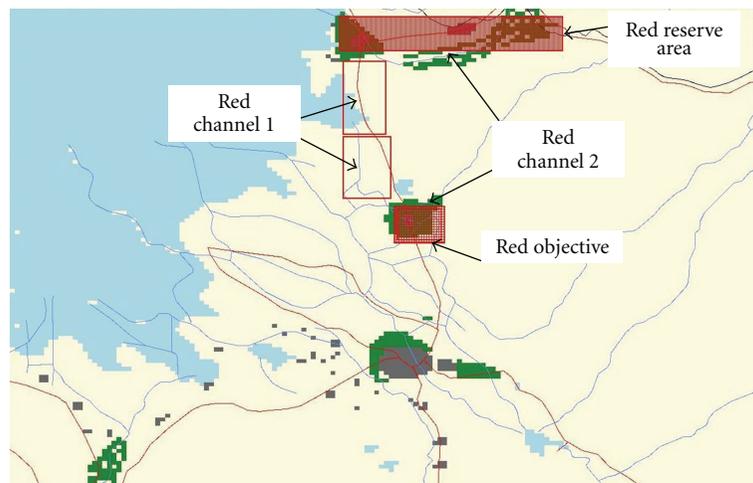


FIGURE 7: Red areas of operation (“channels”) are shown, within the context of Red intent (the Red Objective).

a number of variations of the Deliberate Planner making a number of different assumptions about the fitness function. In the Deliberate Planner, the broad movement of the forces on the ground is task organised into ‘channels’ or areas of operation, which head towards objectives (such as an area of ground to be attacked or a capital city to be defended). These are options which the Deliberate Planner can use in its consideration of how to deploy the force, and forces can be moved between channels as the scenario progresses, as part of the Plan Supervision and Repair process. In our future scenario, there are two Blue channels (Figure 6) and two Red channels (Figure 7). Red are initially static with Blue moving towards their objective. In order to make a fair comparison, both the players in the wargame, and the closed-form simulation, started with the same information from sensors and intelligence reports and had the same initial appreciation of the battlespace in terms of movement and key areas of ground. Thus, for example, the initial Recognised Picture available to a Blue commander in the SE version of the scenario had the same

information content as the Recognised Picture available to the planner representing that commander in the closed-form constructive simulation version. Of course this could diverge as the scenario unfolded, depending on the choices made subsequently either in the SE or the closed-form version. It was also assumed in each case that the Unmanned Air Vehicles (UAV) deployed as sensors could not be shot down, in order that a reasonable level of situational awareness could be maintained within the Recognised Picture and that this factor (i.e., loss of sensor input) would not greatly influence the plan created. For our comparison, the planner was run with a cautious command style (i.e., a maximin payoff function was assumed for Blue, as part of the evaluation of his plan fitness). A higher weighting in the fitness function was also given to the impact of Blue’s plan on Red forces than the impact of Blue’s plan on Blue forces. The comparison was made on the basis of a Turing Test—could an expert military commander tell the difference between a plan generated by human players and a plan generated by the computer algorithms?

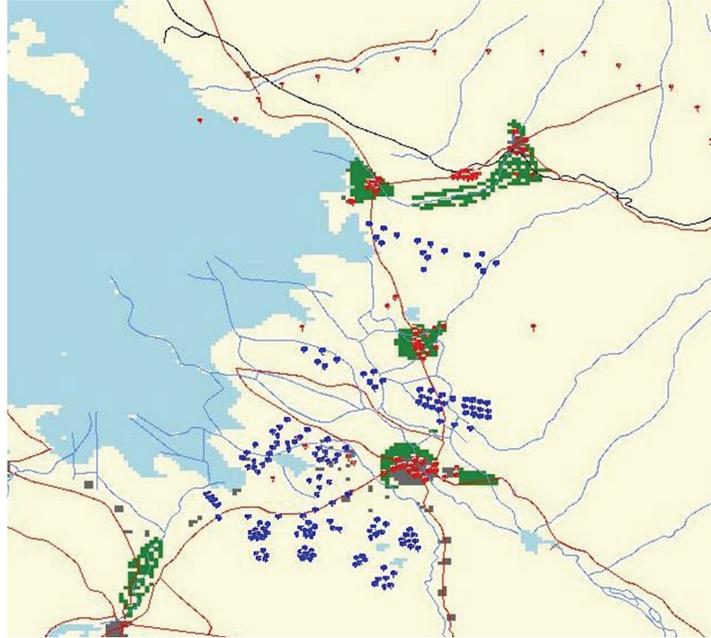


FIGURE 8: Blue and Red initial force deployment locations. Note that some Blue “screening” forces are initially located well to the north of the Red Objective in an aggressive posture. Some Red forces are also initially deployed to the south of their Objective, in an urban area.

Figure 8 shows the initial deployment of the forces (the same for both the wargame and the closed-form simulation, again to ensure fairness of comparison).

6. Comparison of the Simulation Model Algorithm with the Wargame

6.1. Sensor Tasking. When the Deliberate Planner algorithm is initialised, it allocates airborne unmanned air vehicle sensors (UAVs) to the first zones on the channels. Data is used to define the list of sensors allocated to each channel, as well as how many should be used on that channel at any one time.

The output log from the closed-form simulation showed that both the initial sensor tasking and subsequent sensor tasking took place in the model, with information from these sensors influencing the Deliberate Planner. The initial sensor tasking can be seen visually in Figure 9.

6.2. Unit Tasking. A plan is generated either when the sensors have searched all of the zones three times or when a user defined trigger time has been reached. In terms of the simulation run, the plan generated was triggered by the user-defined time. Prior to the generation of the plan, information is supplied to the Deliberate Planner to allow it to build up its Recognized Picture. An idea of the type of information available in completing this situational assessment can be seen in the two-Brigade perception screenshots at Figure 10 showing the initial assessment made by the planner and then a more refined assessment as further sensor information is taken into account by the algorithms.

The acquired locations of the enemy force as well as the location of own force units are used to generate the Recognized Picture. It should be noted that acquisitions made outside of the defined zones are not passed into the Deliberate Planner. The algorithm only considers those acquisitions within the zones where it maintains its Recognized Picture, thus the plan is defined on the basis of the planner’s *perception*.

Figure 11 shows the execution of the plan within the simulation following the dissemination of orders to the forces. This higher level plan is a “left hook” by Blue forces (similar to the plan adopted by the Allies in operation Desert Storm), bypassing concentrations of Red force in order to achieve Blue’s objectives and intent in a timely way. A small allocation of Blue force is also directed towards the Red perceived objective in order to “fix” Red forces in place.

7. Discussion

The plan that was generated sent the majority of units along the “left hook,” with only two company-sized units being tasked down the second channel to “fix” the Red forces. Since the planner is clearly trying to reach the objective as quickly and with as few casualties on its side as possible, the plan is militarily credible, and the military assessment was that it could have been generated by a human commander. By choosing the left hook, the main enemy dispositions in the two urban areas are bypassed so that the objective can be reached through the least cost path. Bypassing urban areas rather than clearing them is an accepted tactic in order to maintain tempo, but the enemy left behind must be fixed or at the very least screened to provide intelligence on enemy

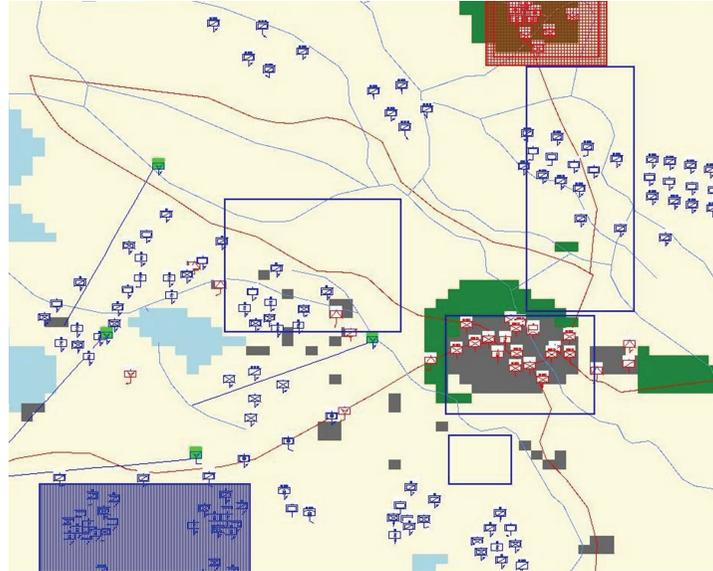


FIGURE 9: Initial sensor tasking and deployment. The slanting lines show the initial flight patterns of the unmanned air vehicle (UAV) sensors, and the open rectangular boxes illustrate typical search areas to which they are directed. The dark blue box is the Blue force reserve area. The dark red box is the Red force Objective.

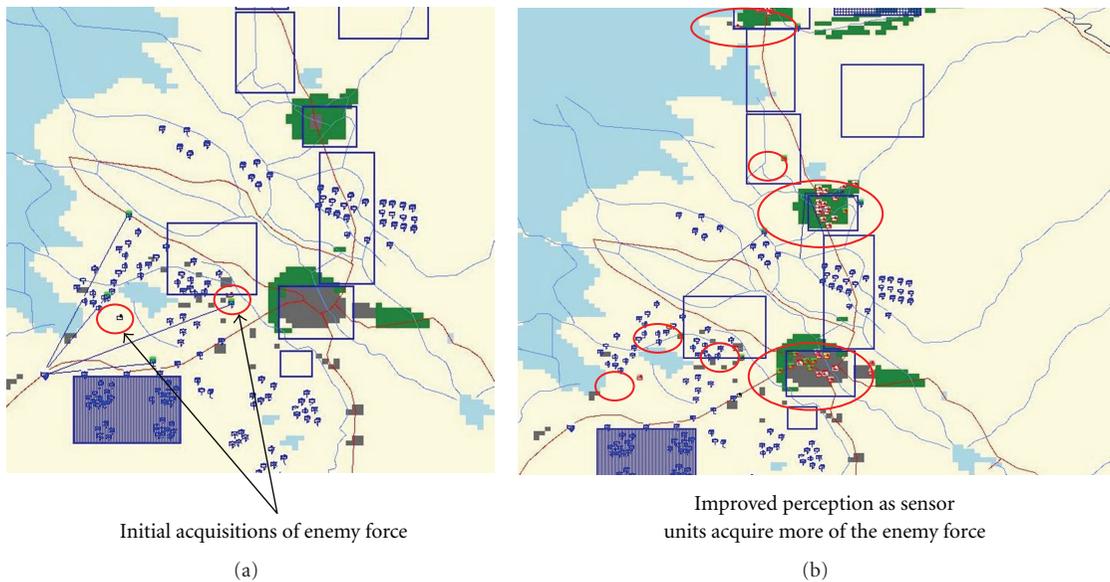


FIGURE 10: The Brigade-Recognized Picture evolution following additional sensor-based information. In both illustrations, the “ground truth” deployment of the Blue forces is shown. In the initial stages of the simulation, a few indicators of Red perceived force locations are reported, shown by the two circles in the left hand illustration. As the situation evolves over simulation time, the sensors pick up more of the perceived Red locations, shown in the right hand illustration. These changing assessments of Red layout of forces form the basis of the evolution of the Recognized Picture and the Plan Repair Process.

movement. In the orders generated by the planner in the simulation, the allocation of Blue units to these areas would be insufficient to conduct this without support from other assets such as Attack Helicopters or Indirect Fire which were not considered here but would be in reality. Variations using the planner indicated that when using the criterion of pessimism (maximin), the planner tended, in general, to

form conservative plans with relatively low risk. In contrast, when using the criterion of optimism (maximax), the plans tended to be high risk. The results of these variations actually carried out in the CLARION model—see Figure 1. were in accord with military judgement. As a consequence, a third criterion, intermediate between these two, was formed as an alternative command “style.”

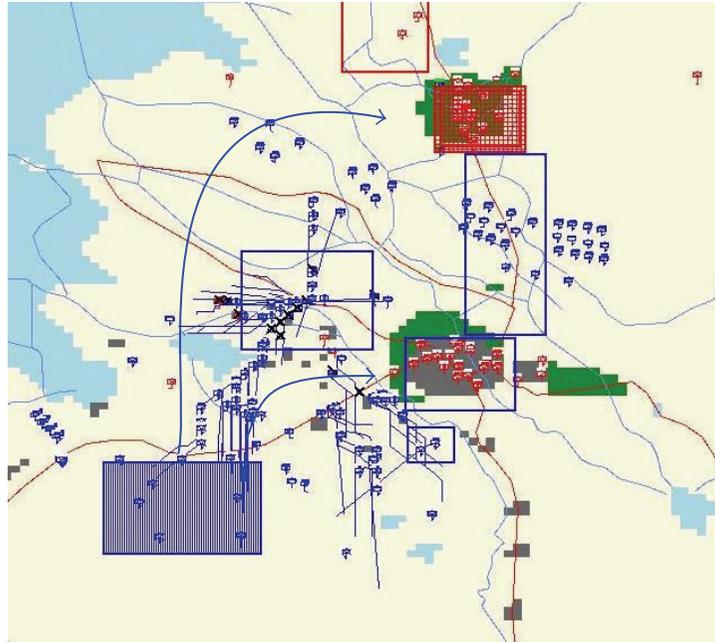


FIGURE 11: The higher level plan generated and implemented by the simulation. The main Blue force performs a “left hook” manoeuvre, while a small Blue “fixing” force holds the forward positioned Red forces in the urban area.

8. Further Developments

We have demonstrated that higher level planning can be carried out using genetic algorithms and produces militarily credible plans. This approach is being exploited further within the UK in current model developments. In one of our simulation models (CLARION), we are developing a Mission Planner based on the same genetic algorithm approach employed in Deliberate Planning. The approach being constructed is that each unit develops a local plan using Rapid Planning. These resultant “missions” or course of action choices are then coordinated within an area by the Mission Planner. Meanwhile, the Deliberate Planning algorithms deal with the larger-scale allocation of forces to areas of operation.

References

- [1] B. Taylor and A. Lane, “Development of a novel family of military campaign simulation models,” *Journal of the Operational Research Society*, vol. 55, no. 4, pp. 333–339, 2004.
- [2] J. Moffat, “Representing the command and control process in simulation models of conflict,” *Journal of the Operational Research Society*, vol. 51, no. 4, pp. 431–439, 2000.
- [3] J. Moffat, I. Campbell, and P. Glover, “Validation of the mission-based approach to representing command and control in simulation models of conflict,” *Journal of the Operational Research Society*, vol. 55, no. 4, pp. 340–349, 2004.
- [4] A. Robinson and S. Wright, “The Wargame Infrastructure and Simulation Environment (WISE),” in *Proceedings of the Operational Research Society Simulation Workshop*, Birmingham University, UK, 2002.
- [5] P. Pearce, A. Robinson, and S. Wright, “The Wargame Infrastructure and Simulation Environment (WISE),” in *Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES ’03)*, Oxford, UK, 2003.
- [6] J. Moffat, *Simulation Modelling for the Information Age: The Quest for Agility*, CCRP, US DoD, Washington DC, USA, 2010.
- [7] J. Moffat, *Command and Control in the Information Age: Representing Its Impact*, The Stationery Office, London, UK, 2002.
- [8] J. Holland, “Genetic algorithms and the optimal allocation of trials,” *SIAM Journal of Computing*, vol. 2, no. 2, pp. 88–105, 1973.
- [9] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [10] D. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading, Mass, USA, USA, 1989.
- [11] D. Lawrence, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
- [12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, Germany, 1992.
- [13] J. Grefenstette, *Genetic Algorithms for Machine Learning*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
- [14] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 2001.
- [15] W. D. Hillis, “Co-evolving parasites improve simulated evolution as an optimization procedure,” *Physica D*, vol. 42, no. 1–3, pp. 228–234, 1990.
- [16] D. Rowland, *The Stress of Battle: Quantifying Human Performance in Combat*, The Stationery Office, London, UK, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

