Human Genome Behaviour: A Powerful Mechanism for Optimizing the use of Space Technology in Surveying Networks Design

Hussain Aziz SALEH & Frank VANDEN BERGHEN
Institut de Recherches Interdisciplinaires et de Developpements en Intelligence Artificielle,
IRIDIA, CP 194/6, Universite' Libre de Bruxelles,
Avenue Franklin Roosevelt 50, 1050 Bruxelles, Belgium.

Abstract

An approach based on the behaviour of human genome is developed to efficiently provide a general framework for optimizing the use of space technology in surveying networks design. The developed approach attempts to apply the successful self-organizing principles based upon the biological evolution to artificial intelligence. It mimics the phenomena of natural selection observed in nature to achieve its goals by continuously adapting a population of candidate solutions and improving its performance over successive generations. The goal of adaptation is to find the best solution that optimizes the design of a surveying network based on the use of satellite observations? This network can be defined as a set of stations, co-ordinated by a series of sessions formed by placing receivers on the stations. The problem is to search for the best order for observing these sessions to give the best observation schedule at minimum cost. The obtained results prove the effectiveness of the developed technique in term of solution quality and computational efforts.

Key words: Global Positioning System GPS, Metaheuristics, Artificial Intelligence AI, Genetic Algorithms GAs.

1. INTRODUCTION

Considerable work has been done through research, field trial, and industry validation to develop the use of satellite technology for real life applications (Leick, 2004). This paper deals with the use of Global Positioning System (GPS) to establish surveying networks. Today, Artificial Intelligence (AI) support is building in satellite technology for the innovative, and leading researches to make the transition from evaluation to adoption (Saleh, 2003). Genetic Algorithms (GAs), which are based on the biological evolution, have become an interesting tool for optimizing many complex real-world problems (Goldberg, 1989). To explain the importance of GAs with regards to the satellite surveying, consider the simple four stations network of Figure 1 where surveyors with their receivers must visit these stations according to the initial observation schedule of Table 1. The surveyor knows the traveling time between stations and wishes to minimize the total time elapsed. The challenge is that "*in what an effective order should the surveyor visit the stations*?" To try every possible route might be feasible in a small network, but search domains for some solutions are very exhaustive for large networks (Dare and Saleh, 2000). In this case, the GA metaheuristic technique can be used to find a better solution in much less time. Generally, a metaheuristic technique is an iterative procedure for quickly and efficiently identifying a high quality solution for complex real world problems (Osman and Kelly, 1992). More precisely, the aim

of this technique is to search and determine the most suitable solution for optimizing an objective function (cost, accuracy, time, etc.) over a discrete set of feasible solutions to a problem to be optimized as shown in Figure 2. In this paper, the GA metaheuristic technique has been used to minimize the time of observing GPS surveying networks. Although it probably will not find an optimal solution, but it can quickly find a near optimal solution which is practically satisfy the essential requirements for designing a GPS surveying network.
GAs were designed to simulate processes in natural systems necessary for evolution; especially these follow the principles first laid by Charles Darwin of "survival of the fittest" (Holland, 1970s). In nature it is competition among individuals for scanty resources that results in the fittest individuals dominating over the weaker ones. To effectively use GAs, a biological representation (as a genome or chromosome) of a solution to an optimized problem is needed. Then, GAs create a population of solutions and apply genetic operators (e.g., mutation and crossover) to evolve the solutions over successive generations in order to find the best one(s). This paper introduces GAs and proposes a modified approach based on the genetic behaviour to search for the best possible schedule to observe a GPS surveying network. Computational results are reported and concluding remarks are summarized. New research is outlined.

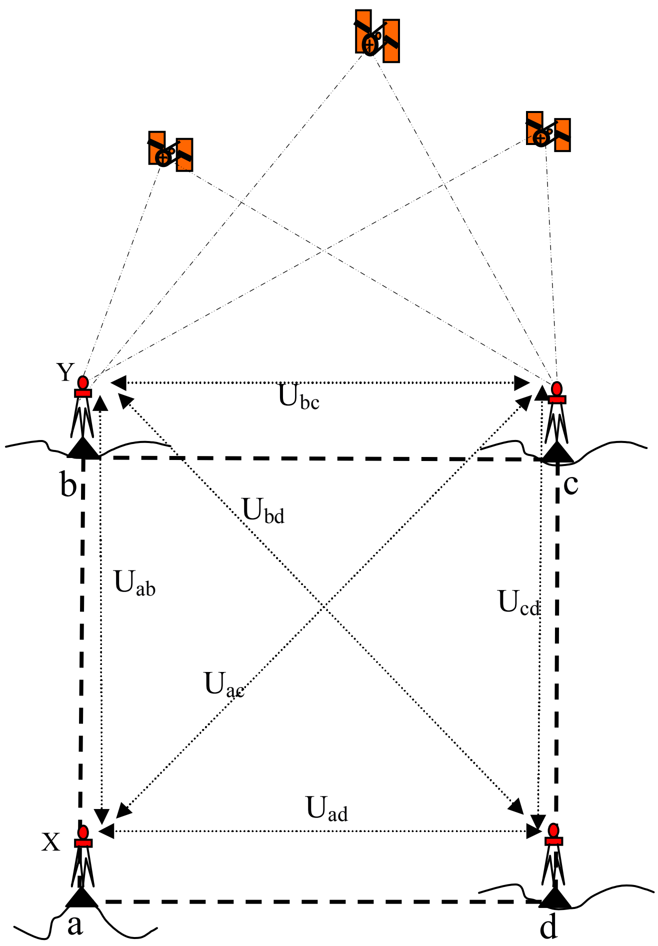

Figure 1. Simple four stations network.

Session	Receiver X at station	Receiver Y at station	Cost
U_{ab}	a	b	0
U_{ac}	a	c	C_{bc}
U_{ad}	a	d	C_{cd}
U_{cb}	c	b	$C_{ac}+C_{db}$
U_{cd}	c	d	C_{bd}
U_{bd}	b	d	C_{cb}
The total cost C(V)			$\sum C_{ij}$

Table 1. The initial observation schedule using 2 receivers.

2. FORMULATION OF THE PROBLEM

In this paper, the modified GPS-GA technique constructs from scratch an initial observation schedule and improves its quality by reducing its total cost over successive iterations. This total cost represents the sum of the cost of moving receivers between GPS stations. For example, as shown in Figure 1 and Table 1, at the beginning of the survey there will be no movements for the receivers when observing the initial session U_{ab} between station *a* and station *b* (i.e., the cost of observing session U_{ab} is 0). To resume the survey and observe session U_{ac}, receiver Y is moved from station *b* to station *c*, while receiver X remains located at station *a* (i.e., there is only one movement). The cost of observing session U_{ac} is obtained by C_{bc} the cost associated with moving receiver Y from station *b* to station *c*. To observe session U_{ad}, receiver Y is moved from station *c* to station *d*, while receiver X remains located at station *a* (i.e., there is only one movement). The cost of observing session U_{ad} is obtained by C_{cd} the cost associated with moving receiver Y from station *c* to station *d*. To observe session U_{cb}, receiver X is moved from station *a* to station *c* and at the same time receiver Y is moved from station *d* to station *b* (i.e., there are two movements). Therefore, the total cost of observing session U_{cb} is obtained by C_{ac} the cost associated with moving receiver X from station *a* to station *c* and by C_{db} the cost associated with moving receiver Y from station *d* to station *b*. Similar procedure will be applied to the sessions U_{cd} and U_{bd} to obtain their observation costs C_{bd} and C_{cb} respectively. At the end of the survey, the total cost C_{ij} to observe all the required sessions will be calculated and minimized. To formulate the above survey as an optimization problem, the GPS surveying network problem within the frame of metaheuristic can briefly defined as follows. A number of receivers (X, Y, etc) are placed at stations (a, b, c, d, etc) to be co-ordinated by determining sessions (ab, ac, dc, etc) between these stations. The problem is to search for the best order in which to consecutively observe these sessions to have the best observation schedule at minimum cost, i.e.,

$$\textit{Minimise} \ : \mathrm{C(V)} = \sum \mathrm{C_{ij}}$$

where

C(V) : the total cost of a feasible schedule V.
$\sum C_{ij}$: the total cost of carrying out the whole survey and *i*, *j* represent stations of the network.

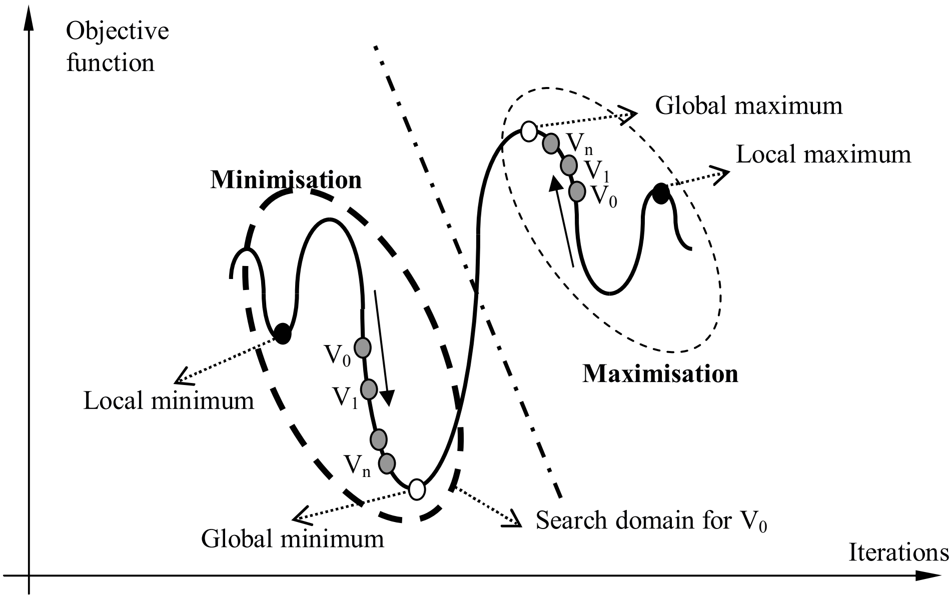

Figure 2. Global and local optima concept for the search domain of the initial schedule V_0.

3. THE REAL BEHAVIOUR OF HUMAN GENES

All living organisms consist of cells where each cell has the same set of chromosomes (or genomes). These chromosomes are strings of genes for defining species which serve as a model for the whole human organism. These species are blocks of DNA that control a hereditary characteristic. A gene carries biological information in a form that must be copied and transmitted from each cell to all its progeny. A gene pool is the collection of all alleles in a particular population (set of chromosomes). A particular set of genes describing an individual is called a genotype which contains the information required to construct an organism. For example, each gene which is a string of bits encodes a particular protein or basically trait (e.g., color of eyes). Possible setting for a trait (e.g., blue, brown, yellow, etc) are called alleles (the set of available options for the next generation). The natural evolution is done through the process of the recombination which takes two original genotypes and produces new ones by mixing their gene values. This evolutionary process can be carried out as follows; from an initial generation; select individuals; breed them; mutate their genes using genetic operators; and insert them into the next generation (Renders and Flasse, 1996). Figure 3 shows the basic pseudo-code procedure to generate a population, while Figure 4 illustrates this procedure graphically.

The genetic operators (selection and recombination) are used to alter the composition of the population. Selection is done on the basis of relative fitness and it probabilistically culls from the population those designs that have relatively low fitness. Recombination, which consists of mutation and crossover, imitates sexual reproduction by combing genes from two existing chromosomes (parents) to form two new chromosomes (children) that have a high probability of having better fitness than their parents. Crossover, as an assorting operator, is applied with high probability to allow information exchange between candidate solutions. In contrast, mutation, as in natural systems, is a very low probability operator and just flips a specific bit in the solution itself to allow new areas of the search domain to be explored. Figure 4 illustrates how crossover mixes information from two parent strings and produces offspring made up of parts from both parents, while mutation flip the parts of the solution itself (Chelouah and Siarry, 2000) and (Jaszkiewicz and Kominek, 2003). In summary, selection concentrates the search process in promising areas of the search domain of the solution, while crossover and mutation provide new points in this domain to be evaluated (solutions with further improvements).

```
create a random population n (generation =0);
repeat {
   repeat {
       father, mother = select (n (generation));
       child = recombine (mutate (father), mutate (mother));
       add child to n (generation+1);
    } until n (generation+1) is full;
    generation= generation+1;
}until terminated;
```

Figure 3. The basic pseudo-code procedure for the classical GA approach.

As can be seen, GA technique will converge over successive iterations towards the global (or near global) optimum via the genetic operations. It maintains a population of chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan. Consequently, highly fit solutions are given more opportunities to reproduce, so that offspring inherit characteristics from each parent. Each new successive generation will contain more good genes 'partial solutions' than previous generations. In the next section, these concepts of GAs are modified to the GPS surveying network to produce an optimal observation schedule (or close to it).

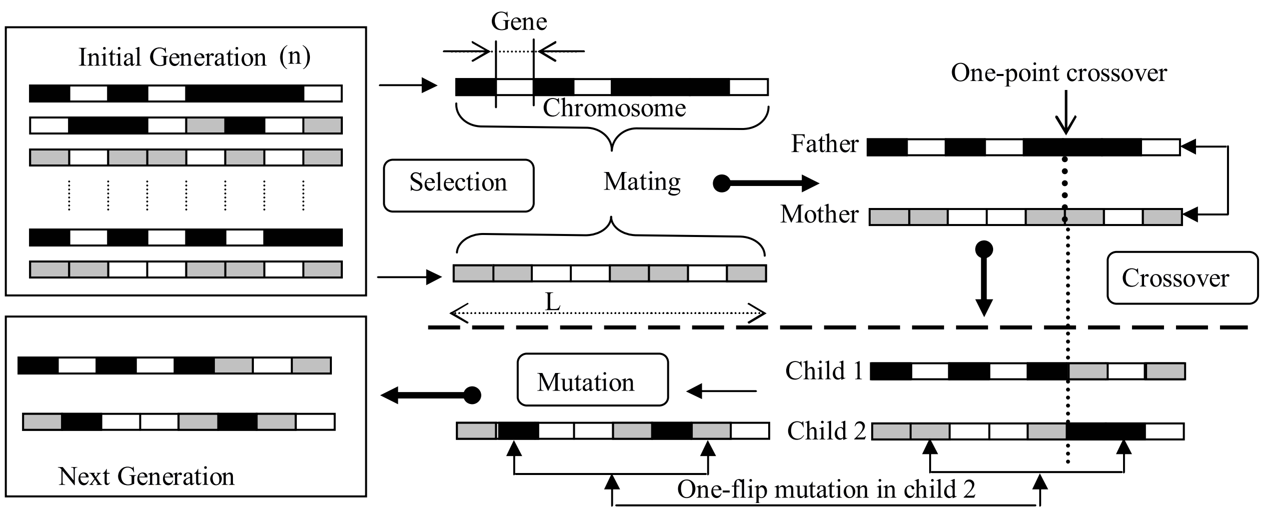

Figure 4. The evolution process for generating a population.

4. HOW THE DEVELOPED GPS-GA TECHNIQUE WORKS?

The most three important aspects of using genetic ideas in satellite surveying are: definition and implementation of the genetic representation, definition of the objective function, and definition and implementation of the genetic operators. The proposed GPS-GA technique is a computer model in which a population of candidate observation schedules to a GPS surveying network are stochastically selected, recombined, mutated, and then either eliminated or retained based on their relative fitness. This can be carried out through four basic steps as illustrated in Figure 5. In the *first step*, the initial population of chromosomes (schedules) is created either randomly or by perturbing an input of the chromosome. In the *second step*, the evaluation of the fitness (cost) is computed. The exploitation phase or natural selection is carried out in the *third step*. In this step, the chromosomes with the largest fitness scores are placed one or more times into a mating subset in a semi-random fashion, while chromosomes with low fitness scores are removed from the population. In the *fourth step*, the exploration phase consists of the recombination and mutation operators.

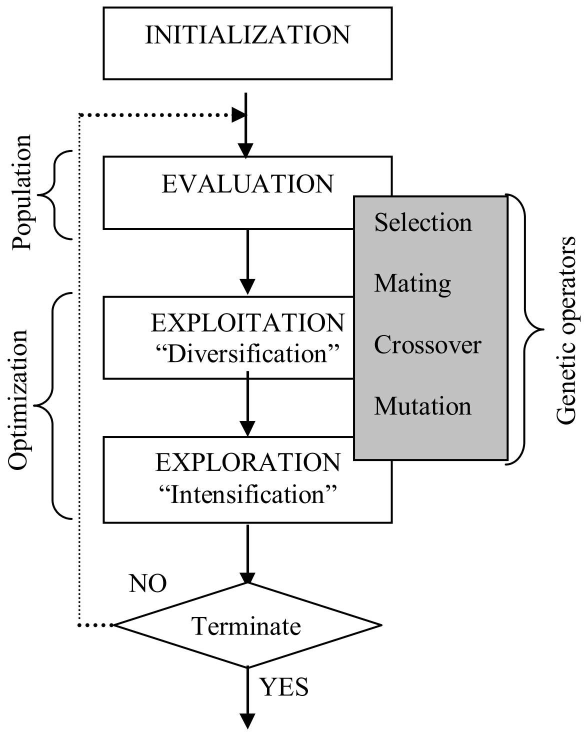

Figure 5. The general flowchart of the GAs.

These four steps of the Figure 6 can be applied to the evolution process of Figure 4 as follows. Two chromosomes with length L (schedules with defined number of sessions) from the initial population are randomly selected to be mated (reproduced). The probability that these chromosomes are mated is a user-controlled option and is usually set to a high value (e.g., 0.90). If the parents are allowed to mate, a recombination operator is employed to exchange genes (sessions) between the two parents (old schedules) to produce two children (new better quality schedules). If they are not allowed to mate, the parents are placed into the next generation unchanged. In case the parents are allowed to mate, a crossover point is selected along the chromosome and the genes up to that point are swapped between the two parents. The children then replace the parents in the next generation. A mutation operator is employed to increase the search diversity in the population. The probability that a mutation will occur is another user-controlled option and is usually set to a low value (e.g., 0.01) so that good chromosomes are not destroyed. After the exploration phase is carried out, the population is full of newly created chromosomes and steps two through four of Figure 5 are repeated. This process continues for a fixed number of generations (iterations). One of the features that separate GA approach from other optimization approaches is that it optimizes on a representation of the variables and not the variables themselves. This allows GA technique to be very flexible and to optimize well in situations where the variables are much different from each other.

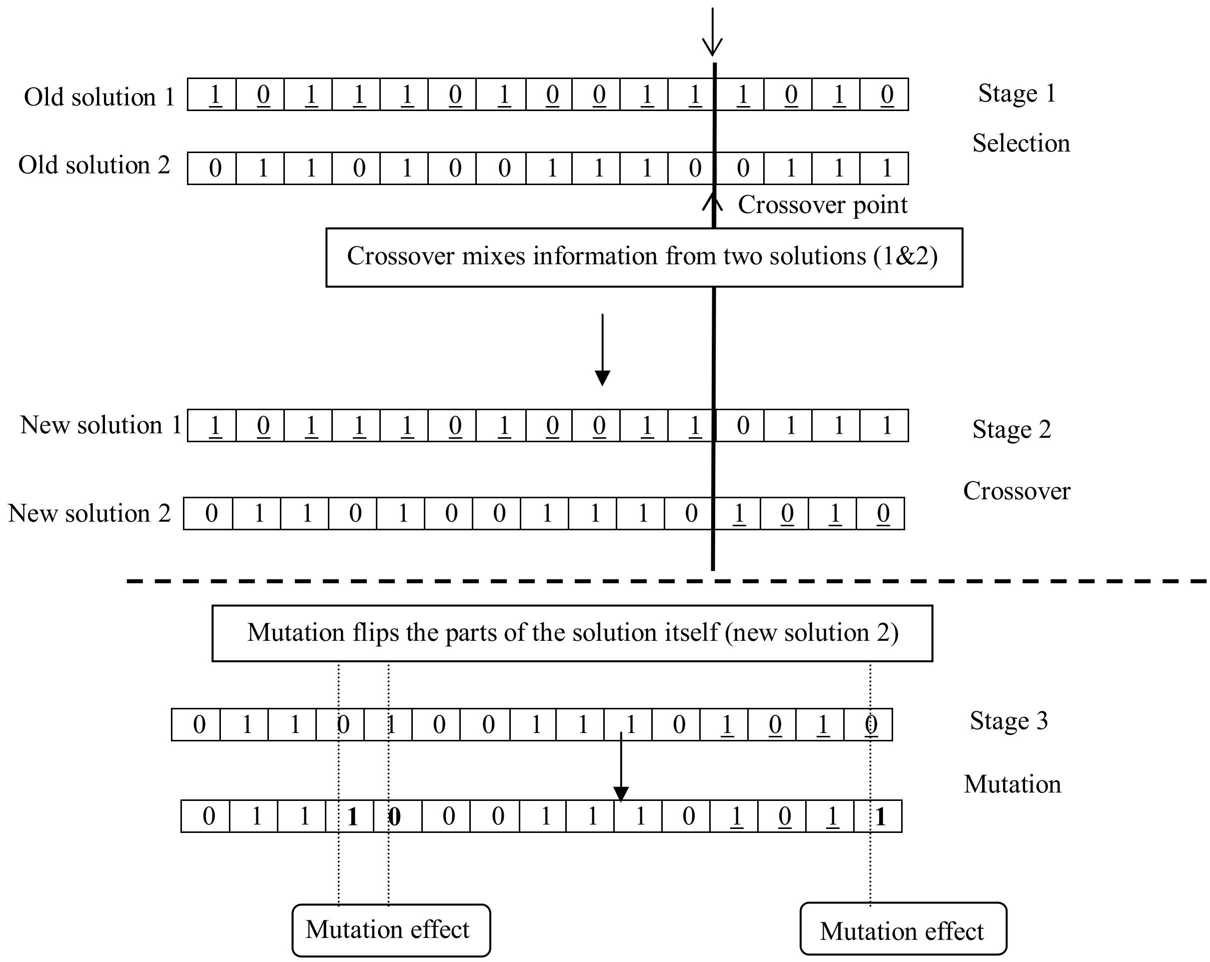

Figure 6. The binary-coded representation of GA technique.

Many researchers have noted that one of the most common reasons that GA technique does not perform well in certain applications is due to the poor choice of representation. The decision of which this representation should be used is based upon the characteristics of the variables being optimized. For example, in certain applications such as wavelength selection in spectroscopy present an easy choice because wavelengths (i.e., genes in the chromosome) are either used (1) or not used (0). Thus, in a binary-coded GA, each chromosome is a vector comprised of zeroes and ones with each bit representing a gene as shown in Figure 6. The GA technique creates a population of solutions then applies crossover and mutation to these solutions to generate new solutions. In the second stage, a *crossover point* will be chosen for each pair and the information after this point will be exchanged between the two solutions of each pair. In the last stage, the mutation adds some effect of *exploration* of the phase-domain to the algorithm by simply changing a one to zero or a zero to a one in the same solution. The generated "offspring" will include some solutions that are better than the original ones. Encouraging the best generated solutions and throwing away the worst ones "only the fittest survive", the original population keeps improving as a whole and this is called "selective pressure". By repeating this process among the better generated solutions, increased improvements will be gained during the successive generations (Michalewicz, 1996).

[I] INITIALISATION & REPRESENTATION:

(A) *FORMULATING* the original cost matrix (time travels between stations):
Insert the total number of stations and the estimated cost for each receiver's move.

(B) *CREATING* the actual cost matrix (time travels between sessions):
Insert the number of receivers and define the sessions to be observed,

(C) *RPERESENTING* a potential solution to the observed network.
Encode the solutions as strings over a finite alphabet.
Each string must represent a complete solution to the observed network.

(D) *DEFINING* the cost function:

(E) *DETERMINING* the structural elements:
Set the population size *n*;
Set the initial populations POP and POP' at times t and t+1 respectively;
Set the f_{min} which is the lowest cost of one schedule through the population POP;
Set the $\boldsymbol{x}_{min}$ which is the schedule that allows obtaining this lowest value of the cost function.

(F) *INITIALIZING* the control parameters:
Set the crossover probability *pCross*;
Set the mutation probability *pMut*;
Set the decreasing law of mutation probability;
Set the stopping criteria.

[II] GENERATION AN INTIAL POPULATION AND ACCEPTANCE STRATEGY

```
while not termination condition do
 (G) DIVERSIFICATION
   generation := generation + 1
   repeat
      Selection
      Crossover(pCross)
      Mutation(pMut)
   until POP′ is full
      Evaluation of POP′
      POP := The new population between (POP, POP′ )
   if (detection of a potential schedule)
   then
 (H) INTENSIFICATION
   "2-opt" Procedure
         end while
```

[III] THE STOPPING RULES:

(I) *TERMINATING* the search process:
Stop if the stopping criterion is satisfied; when
Given number of iterations, OR
Maximum number of iterations allowed without improving the best generated schedule.
OTHERWISE.
Go to (E).

(J) *DECLARING* the output:
Declare the best generated schedule;
Declare the computation time;
END.

Figure 7. The general outline of the developed GPS-GA technique.

The genetic representation of the GPS surveying network problem is a difficult task and the complexity is how to encode the observation schedules according to the GA strategy. The encoding cannot simply be the list of sessions in the order that they have to be observed. Therefore, a sophisticated form of encoding must be used to generate a potential schedule as explained in the following section. After the proper representation of the feasible schedule is carried out, an initial population of this potential schedule is generated and a *fitness function* (or *objective function*) is used to evaluate each schedule in this population (ranking and scaling). The genetic operators are applied to alter the composition of the generated schedules. At each time step (or generation), the algorithm produces a population of possible schedules based on the population from the previous time step. Then, selection functions are used to select a given number of schedules from the current population according to their cost (fitness). The common selection method is roulette wheel selection in which the likelihood of picking a schedule is proportional to the cost of the schedule. The various parameters (population size, probabilities of applying the genetic operators, etc) will be tuned at each generation. The general outline of the developed GPS-GA technique is shown in Figure 7.

5. COMPUTATIONAL RESULTS

This section reports on the computational experience of the GAs technique using GPS networks with known and unknown optimal solutions. The performance of the proposed GA-GPS technique was evaluated with respect to the schedule quality and computational effort. The best known solutions were obtained for relatively small GPS surveying networks using the Traveling Salesman Problem (TSP) algorithm (Dare, 1995). Figure 8 shows a hypothetical network which has an optimal schedule with a cost of 13 minutes. The data set for this network consists of four stations, two receivers and six sessions. Table 2 represents the time travels between stations (i.e., original cost matrix), while Table 3 represents the time travels between sessions (i.e., actual cost matrix). The initial starting schedule V_0 with a cost of 17 minutes was randomly chosen and consisted of the following sessions [V_0 = U_{ab}, U_{ac}, U_{ad}, U_{bd}, U_{cd}, U_{cb}]. The metaheuristic schedule generated by the developed GPS-GA technique for this hypothetical network had the same cost as the known optimal schedule (Chelouah and Saleh, 2003). To illustrate the search methodology of GPS-GA technique, the diversification and intensification phases are performed during the search process of optimizing the initial observation schedule of network of Figure 8 as follows:

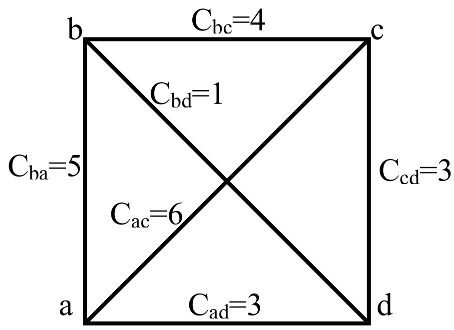

Figure 8. Four-station network (from Dare 1995).

	N_a	N_b	N_c	N_d
N_a	0	5	6	3
N_b	5	0	4	1
N_c	6	4	0	3
N_d	3	1	3	0

Table 2. Original cost matrix for network in Figure 8 (from Dare, 1995).

	U_{ab}	U_{bc}	U_{cd}	U_{da}	U_{ad}	U_{dc}	U_{cb}	U_{ba}	U_{bd}	U_{db}	U_{ac}	U_{ca}
U_{ab}	0	5	6	4	1	4	6	0	5	3	4	6
U_{bc}	4	0	4	6	4	1	0	6	3	2	4	6
U_{cd}	6	2	0	3	6	0	1	3	2	3	6	3
U_{da}	5	6	3	0	0	6	5	1	3	5	6	2
U_{ad}	1	5	6	0	0	3	6	5	5	3	2	6
U_{dc}	3	1	0	6	3	0	2	6	3	2	3	6
U_{cb}	6	0	1	4	6	4	0	4	2	3	6	4
U_{ba}	0	6	4	1	4	6	5	0	3	5	6	4
U_{bd}	4	2	4	3	4	2	4	3	0	0	4	4
U_{db}	3	4	2	4	3	4	2	4	0	0	4	4
U_{ac}	2	5	6	6	3	3	6	6	5	3	0	0
U_{ca}	6	6	3	3	6	6	5	2	3	5	0	0

Table 3 Actual cost matrix for network in Figure 8 (from Dare, 1995).

5.1. Diversification Phase consists of the following stages:

5.1.1 Initialization of the population

In the beginning, an initial schedule V_0 is selected and the iterations counter is initially zero, indicating no moves are performed at the beginning of the search process.

$$V_0=(U_{ab})\text{--}[4]\text{-->}(U_{ac})\text{--}[3]\text{-->}(U_{ad})\text{--}[5]\text{-->}(U_{bd})\text{--}[4]\text{-->}(U_{cd})\text{--}[1]\text{-->}(U_{cb}) = \{17\}$$

To produce new schedules and generate the population of V_0, the sessions are randomly exchanged by M permutations (i.e., number of sessions M=6) as shown in Table 4.

Generated Schedules	order of sessions	cost C(V)
V_0	U_{ab} U_{ac} U_{ad} U_{bd} U_{cd} U_{cb}	17
V_1	U_{ac} U_{ab} U_{ad} U_{bd} U_{cd} U_{cb}	13
V_2	U_{ab} U_{ac} U_{ad} U_{cb} U_{cd} U_{bd}	16
V_3	U_{bd} U_{ac} U_{ad} U_{ab} U_{cd} U_{cb}	15
V_4	U_{ad} U_{ac} U_{ab} U_{bd} U_{cd} U_{cb}	14
V_5	U_{cd} U_{ac} U_{ad} U_{bd} U_{ab} U_{cb}	24
V_6	U_{ab} U_{bd} U_{ad} U_{ac} U_{cd} U_{cb}	18
V_7	U_{cb} U_{ac} U_{ad} U_{bd} U_{cd} U_{ab}	24
V_8	U_{ab} U_{ac} U_{bd} U_{ad} U_{cd} U_{cb}	20
V_9	U_{ab} U_{cd} U_{ad} U_{bd} U_{ac} U_{cb}	27
V_{10}	U_{ab} U_{cb} U_{ad} U_{bd} U_{cd} U_{ac}	27
V_{11}	U_{ab} U_{ac} U_{cd} U_{bd} U_{ad} U_{cb}	22
V_{12}	U_{ab} U_{ac} U_{ad} U_{cd} U_{bd} U_{cb}	19
V_{13}	U_{ab} U_{ac} U_{cb} U_{bd} U_{cd} U_{ad}	22

Table 4. The population size generated from the initial schedule.

5.1.2 Selection

The selection operation is based on the concept of the roulette-wheel selection and can be explained as follows:

Step 1: calculating the objective function (cost value) for each schedule and detecting the best values. For example, the first four schedules were selected from Table 4 as follows:

$f(V_1) = 13$ (i.e., the cost for V_1)
$f(V_2) = 16$
$f(V_3) = 15$
$f(V_4) = 14$

Step 2: calculating the "fitness value" which is the difference between the cost of the cheapest schedule and the other schedule's values $\{Fit(V_i) = Max(f) - f(V_i)\}$. The fitness value for each schedule is replaced by the cumulated fitness which is obtained by adding its fitness to the previous values. In this method the function fitness *Fit* of a schedule *i* depends only on the value of its objective function *f* and on the maximum value of the cost function in the population (Max (f) = 16). For example;

$Fit(V_1) = Max(f) - f(V_1) = 16-13 = 3$ (now the new fitness value is 3 for the V_1)
$Fit(V_2) = Max(f) - f(V_2) = 16-16 = 0$
$Fit(V_3) = Max(f) - f(V_3) = 16-15 = 1$
$Fit(V_4) = Max(f) - f(V_4) = 16-14 = 2$

Step 3: calculating the selection probability by mapping the schedules to contiguous segments of a line. Each schedule segment is equal in size to the associated cumulated fitness. Then, a random number is generated and the schedule whose segment spans this number is selected. This process will be repeated until the desired number of schedules is obtained. Suppose that the resultant population is composed of $n=4$ schedules and functions of adaptation $Fit(j)$ are calculated, the probability P_i of selection a schedule *i* is given by the following expression:

$$p_i = \frac{Fit(\mathrm{V_i})}{\sum_{j=1}^{n} Fit(\mathrm{V_j})}$$

Schedule	Fitness function	p_i
V_1	3	3/6 = 0.5
V_2	0	0/6 = 0
V_3	1	1/6 = 0.17
V_4	2	2/6 = 0.33
Total	6	1.00

Table 5. Selection probabilities of the population's schedules generated from V_0.

where $Fit(V_j) = 6$ (the sum of 3 + 0 +1 + 2).

Table 5 gives the probabilities required for selection of each schedule i for a population of four schedules. As can be seen from this table, the schedule V_2 does not have any chance to be retained for the phase of reproduction. This is due to the smallest value of its function objective which results in a null value of "fitness". Each schedule of the present population occupies a sector of the roulette-wheel with surface proportional to the adaptation value of this schedule as shown in Figure 9.

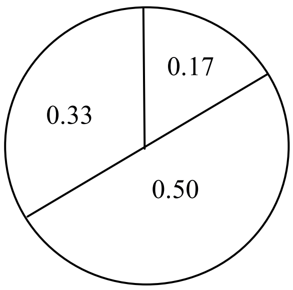

Figure 9. Sectors of the roulette are proportional to the adaptation of the schedules.

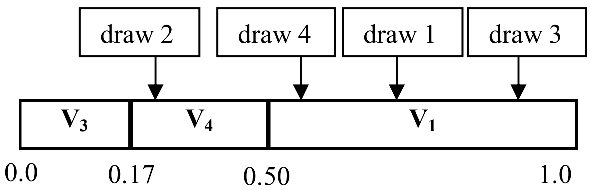

Figure 11. Representation by segments using roulette wheel selection.

The roulette wheel selection is directly derived from the proportional selection in which the schedules are represented by segments equal to the value of their "fitness" (portion of circle) as shown in Figure 10. To select a schedule which is likely to be reproduced, a number ranging between 0 and 1 is randomly drawn and then the fitness value is located. To generate a population of four schedules, four draws will be carried out as illustrated in Figure 10. For example, the schedule V_1 with best adapted value was drawn 3 times, whereas the schedule V_3 with poor adapted value was not retained for the phase of reproduction.

5.1.3. Crossover

The complexity of applying the GA strategy to the GPS surveying network problem is how to encode the observation schedules? This complexity is illustrated in the example of Table 6 in which the crossover operation will not successfully work and this can be explained as follows. In the old schedules (parents 1 & 2), the crossover point is located after the second session. All sessions in parent 1 (the first old schedule) which is located before the crossover point have to be copied in the same order in the child 1 (the first new generated schedule). Then, all sessions which are located after the crossover point in the parent 2 (the second old schedule) have to be put in the same order in the child 1. The opposite procedure has to be applied to child 2 (the second new generated schedule). As can be seen in the child 1 of Table 6 the session U_{ab} was repeated twice, while the session U_{ad} was missing. The same problem was occurred for the child 2 in which the session U_{ad} was duplicated, while the session U_{ab} was missing. To generate a potential schedule without missing or duplicating any session to be observed, a sophisticated form of encoding was used as follows. This form of encoding should ensure that the selected sessions to be observed are strictly preserved from the parents (old schedules) to the children (new generated schedules) during the generation process. This can be carried out by re-copying each of the four sessions located after the crossing point of the second parent in the new generated child 1, except if they have already existed. In this case, the child 1 will be completed without missing or repeating any session as shown in Table 7 (i.e., the child 1 is a new schedule which contains all the selected sessions to be observed, but in different order). In this table, the same process was applied to generate the child 2.

Parent 1	U_{ac} U_{ab} ⋮ $\mathbf{U_{ad}}$ $\mathbf{U_{bd}}$ $\mathbf{U_{cd}}$ $\mathbf{U_{cb}}$
Parent 2	U_{ad} U_{ac} ⋮ $\mathbf{U_{ab}}$ $\mathbf{U_{bd}}$ $\mathbf{U_{cd}}$ $\mathbf{U_{cb}}$
Child 1	U_{ac} $\boldsymbol{U_{ab}}$ $\boldsymbol{U_{ab}}$ U_{bd} U_{cd} U_{cb}
Child 2	$\boldsymbol{U_{ad}}$ U_{ac} $\boldsymbol{U_{ad}}$ U_{bd} U_{cd} U_{cb}

Table 6. Result of the random crossover.

Parent 1	U_{ac} U_{ab} U_{ad} U_{bd} U_{cd} U_{cb}
Parent 2	U_{ad} U_{ac} U_{ab} U_{bd} U_{cd} U_{cb}
Child 1	U_{ac} U_{ab} U_{bd} U_{cd} U_{cb} U_{ad}
Child 2	U_{ad} U_{ac} U_{bd} U_{cd} U_{cb} U_{ab}

Table 7. Correction of the crossover operation.

5.1.4. Mutation

The rule of mutation operator is to further diverse the search domain of a schedule by selecting a random session and changing its position in this schedule. Examples of mutation in case of the GPS surveying network problem are as follows:

1) Exchange between two sessions chosen randomly (e.g., the first and the third sessions in the schedule V_0).

- Before exchange $V_0 = U_{ab}, U_{ac}, U_{ad}, U_{bd}, U_{cd}, U_{cb,}$
- After exchange $V_0' = \boldsymbol{U_{ad}}, U_{ac}, \boldsymbol{U_{ab}}, U_{bd}, U_{cd}, U_{cb,}$

2) Replacement of a connection between two sessions (e.g., the connection between the first session and the nearest session as shown in Figure 1). For example, the session U_{ab} is close to session U_{ac} and the use of mutation operator can be illustrated as follows:

- Before exchange $V_0 = U_{ab}, U_{ac}, U_{ad}, U_{bd}, U_{cd}, U_{cb,}$
- After exchange $V_0' = U_{ac}, U_{ab}, U_{ad}, U_{bd}, U_{cd}, U_{cb,}$

As can be seen, we are not actually calculating a solution to the problem being treated; we are merely selecting and encouraging the best emerging ones after certain random operations. This is why we actually do not need to know how to solve the problem when using GA technique. We just have to be able to evaluate the quality of the generated solutions at each new iteration by creating a proper "fitness function"; nothing else! This means the implementation of GA technique is easy and can rely on a problem-independent "engine", requiring little problem-related work. The evolution and selection process is problem-independent, while only the fitness function and one that decodes the chromosomes into a readable form are problem-specific.

5.2. Intensification phase

The GPS-GA technique adopts the intensification phase around a promising schedule by slightly changing the order of the sessions to be observed. This can be done by implementing the simple and relatively effective 2-opt procedure as follows. After a potential schedule is generated during *diversification phase*, a simple transformation based on local search method is applied to this schedule (swapping between neighboring sessions to generate all the possible schedules). In intensification phase, it is important to carry out a systematic search by testing all the possible permutations in a schedule and this is depending on the size of the network to be optimized. If the transformation improves the schedule, then this transformation is kept, otherwise has to be changed. The above transformation will be repeated until no further improvement can be obtained from the swapping process between the sessions of the generated schedule. For more details about the 2-opt procedure, the reader is referred to Lin (1965). It is preferable to start the process with a large population to cover the whole schedule search domain. The mutation probability depends on the size of the population and may vary during the optimization process.

Network Information			Initial schedule	GPS-GA technique		Reduction
Network	N	S	V_0	V_{GA}	ET	RRC%
Malta	38	25	1405	940	10	33.10
Seychelles	71	36	994	857	80	13.78

Table 8. The computational results of the GPS-GA technique applied to GPS networks observed in Malta and the Seychelles.

Where
N: number of stations;
S: number of sessions;
V_0: the initial schedule;
V_{GA}: the generated potential schedule by the GPS-GA technique;
ET: the processing time needed to find the potential schedule (in seconds);
RRC= The Relative Reduction of the Cost and given by $[(V_0-V_{GA})/V_0]*100$.

5.3. Large networks

To generalize the developed technique and work with larger networks, two different types and sizes of GPS surveying networks were used. The first network was a triangulation-type of Malta which consists of 38 sessions connecting 25 stations. The initial schedule, with cost of 1405 minutes, represents the actual operating solution which was manually generated using the intuition and experience of the surveyors (Dare, 1994). By implementing the GPS-GA technique, the overall cost of the initial schedule was reduced from 1405 minutes to 940 minutes after 20 seconds. The second network was a linear-type of the Seychelles which consists of 71 sessions connecting 57 stations. The initial schedule, with cost of 994 minutes, represents the actual operating solution which was manually generated using the intuition and experience of the surveyors (Dare, 2000). By implementing the GPS-GA technique, the overall cost of the initial schedule was reduced from 994 minutes to 857 minutes after 80 seconds. Table 8 shows the results obtained by the developed technique for both Malta and Seychelles GPS networks. The tuning of control parameters is an important issue concerning the implementation of metaheuristics. The values obtained for the control parameters are: the initial population size *n* equal to 30, the crossover probability *pCross* equal to 0.9, and the initial mutation *probability pMut* equal to 0.85. The improvements process is terminated when the stopping criterion is met. The adopted stopping criterion is the predefined number of iterations (process time) chosen by the user to find the best possible schedule. Compared to other optimization methods, GA technique is robust, global and easy to apply and less susceptible to getting 'stuck' at local optima, but it tends to be computationally expensive. In this research, the developed GPS-GA technique has been coded in Visual C++.

6. CONCLUSIONS AND FURTHER WORKS

The GA metaheuristic is universal optimization method which requires little knowledge about a problem to be solved. At this point the benefits of the GPS-GA technique has been investigated and evaluated for solving and optimizing the GPS surveying network problem (with the static nature). For the future research, the above model will be dynamically expanded to efficiently provide flexible and computerized procedures for determining high-accuracy models with a higher resolution for other geomatic applications (e.g., modeling atmospheric effects, a global high-accuracy gravity field model, and geographic information systems, etc). Many of the current challenges in optimizing these important applications focus on increasing and improving the functionalities of the established techniques for dealing with more real features, such as temporal and metric capabilities, handling a mixture of continuous variables as well as discrete selections, etc. For example, this model will help in investigating various integrations of the internal and external dynamics involved in the planning, development and inception of the space flight "Mission to Mars". This will provide a clear understanding and thorough analysis of the thermal environment under various atmospheric density and velocity conditions which is fundamental to the future widespread use in mission applications.

Acknowledgments
This research was supported by the Syrian Ministry of Higher Education and by Marie Curie Fellowship awarded to Hussain Saleh. Also, this research is supported by the Metaheuristics Network, a Research Training Network funded by the Improving Human Potential program of the CEC.

The information provided in this paper is the sole responsibility of the authors and does not reflect the European Commission's opinion. The Commission is not responsible for any use that might be made of data appearing in this publication.

REFERENCES

Chelouah, R., Siarry, P., 2000. Continuous Genetic Algorithm: an algorithm for the global optimization of multiminima functions. Journal of Heuristics 6, 191-213.

Chelouah, R., Saleh, H., 2003. Genetic Algorithms for Designing the Global Positioning System (GPS) Networks. In the proceedings of the 7th world multi conference on systemics, sybernetics and informatics SCI 2003, July 27 - 30, Orlando, Florida, USA.

Dare, P., 2000, Seychelles Densification Project'98: The Densification of a Control Network for the Republic of Seychelles by use of the Global Positioning System, Lands and Survey Division, Ministry of Land Use and Habitat, Victoria, Seychelles.

Dare, P., 1994. Project Malta' 93: The Establishment of a New Primary Network for the Republic of Malta by use of the Global Positioning System. Report for Mapping Unit, Planning Directorate, Floriana, Malta.

Dare, P., 1995. Optimal Design of GPS Networks: Operational Procedures, Ph.D. thesis, School of Surveying, University of East London, London, U.K.

Dare, P., Saleh, H., 2000. GPS Network Design: Logistics Solution using Optimal and Near-Optimal Methods. Journal of Geodesy 74, 467- 478.

Goldberg, E. D., 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley.

Holland, J. H., 1986. Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In Machine Learning. San Mateo, CA: Morgan Kaufmann.

Jaszkiewicz, A., Kominek, P., 2003. Genetic local search with distance preserving recombination operator for a vehicle routing problem, *European Journal of Operational Research,* 151(2), 352-364

Leick, A., 2004. GPS Satellite Surveying. 3rd ed. Wiley, Chichester, UK.

Lin, S., 1965. Computer solutions of the traveling salesman problem, The Bell System Technical Journal 44, 2245-2269.

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag Heidelberg.

Osman, I., Kelly, J., 1992. Meta-heuristics: An Overview, in: META-HEURISTICS: Theory & Applications. Kluwer academic publishers.

Renders, M., Flasse, P., 1996. Hybrid Method Using Genetic Algorithms for the Global Optimization, IEEE Transactions on Systems, Man, and Cybernetics, 26, 243-258.

Saleh, H. A., 2003. "An Artificial Intelligent design for GPS Surveying Networks". *GPS Solutions*, 7 (2): 101-108.