GENETIC ALGORITHM FOR DYNAMIC TASK ALLOCATION OF MULTI AUTONOMOUS UNMANNED AIR VEHICLES

Jamal A. F. Azzam
M.Sc.Electrical Engineering,
Portsaid Shipyard, Suez Canal Authority.

Prof.Dr. Hassan El_Deip
Electrical Engineering Dpt.,
Facaulity of Engineering, Suez Canal University.

Prof.Dr. Soliaman M.Sharaf
Power And Machines Engineering Dpt.,
Facaulity of Engineering, Helwan,
Helwan Uniirsty

توزيع المهام على الطائرات بدون طيار فى الظروف المتغيرة باستخدام برنامج يماثل الجينات الوراثية

للطائرات بدون طيار UAV تطبيقات عسكرية مفيدة ومنها الاستطلاع ، البحث عن الاهداف وتدميرها والبحث والانقاذ فى ظروف معادية كالمعارك أو مناطق الكوارث. وحديثاً هناك اهتمام كبير ودراسات عديدة لاستخدام اسراب كبيرة تتعاون فيما بينها لانجاز العديد من المهام، وعلى سبيل المثال مهاجمة اهداف محددة وهذا يتطلب تحديد المهام لكل واحده من هذه الطائرات UAV للاهداف المتباعدة جغرافيا خلال مسارات ممكنة لتقليل الجهود (وقت الطيران – استهلاك الوقود – طول المسار من موقعها الى الهدف و مكانية تعرضها للرادارات المعادية) وكذلك تفادى التهديدات المباشرة من المضادات. ان عملية تحديد المهام Task Allocation (TA) هى واحده من الخطوات الاساسية لاستغلال الامكانات المتاحه لوحدات الاسراب وكذلك لاتاحة التعاون فيما بينها وهى مشكلة معروفة كـ NP Complete Problem اى ان حجم تعقيد حلها يتزايد بصفة متسلسلة مع زيادة عناصرها وهى اعداد UAVs والاهداف ومواقع المضادات وهذا يجعل حلها شاقا جدا من ناحية ويتطلب وقتا طويلا فى حسابها من ناحية اخرى . وهذا يؤدى الى :

- عدم امكانية اجراء هذه الحسابات فى وقت الطيران الحقيقى Real Time .
- عدم امكانية التعاون فيما بين وحدات السرب او الاسراب .
- عدم امكانية كون هذه الطائرات UAVs ذاتية Autonomous . لأنها تتطلب حلاً سريعاً المتغيرات أثناء الطيران ومن تم تعديل سريع للمهام

كل الأبحاث المنشورة فى هذا المجال مبنية على الحالة الساكنة (أى ان مكونات المشكلة من مواقع طائرات أو أهداف ومصادر تهديد محسوبة سلفاً ولا تتغير) وهذا يتنافى مع ظروف الميدان فقد تكون متحركة كما قد يطرأ تهديدات جديدة غيرمعروفة سلفاً.

وفى هذا البحث نقدم خوارزماً جديدا لتحديد المهام فى الظروف المتغيرة وفى وقت الطيران الحقيقى وهو مبنى على مبادىء الخوارزميات الجينية. ويناقش البحث استخدام مبادىء الخوارزميات الجينية فى تطوير حل لمشكلة توزيع المهام (Task Allocation) TA لفريق متعدد من الـ UAVs بطريقة مثالية أو شبه مثالية.

ان نتائج المحاكاه تشير الى ان الخوارزم المقترح هو طريقة مفيدة جدا للتطبيق فى حل مشكلة TA سواء فى الحالة الساكنة أو المتغيرة ويتميز بان التكلفة الكلية للسرب قد تم تصغيرها الى اقصى حد ممكن كما ان لها ميزة قيمة وهى كون الزمن اللازم لحسابها صغير جدا مما يمكن الطائرات من الاستجابة للمتغيرات الميدانية والتعاون فيما بينها وزيادة قدرتها الذاتية فى اتخاذ القرارات واحتساب المسارات وتنفيذها وذلك أثناء الطيران.

ABSTRACT:

Unmanned aerial vehicles (UAVs) have useful military applications, including reconnaissance, search and destroy, search and rescue missions in hazardous environments such as battlefields or disaster areas. Recently, there has been considerable interest in the possibility of using large teams of UAVs functioning cooperatively to accomplish a large number of tasks e.g. attacking targets. However, this requires the assignment of multiple spatially distributed tasks to each UAV along with a feasible path that minimizes effort and avoids threats.

Task Allocation (TA) is one of the core steps to effectively exploit the capabilities of cooperative control of multiple UAV teams. It is an NP-complete problem "non-determistic polynomial time". So the computation can't be implemented in real time, no chance for cooperation among the team members, and no autonomy for these vehicles. The reported papers in this field consider the problem in static condition using different techniques (e.g. auction based, scheduling, linear programming).

In this paper, a new dynamic task allocation algorithm is presented that is based on the principles of genetic algorithm GA). It discusses the adaptation and implementation of the GA search strategy to the task allocation problem in the cooperative control of multiple UAVs. Simulation results indicate that the GA strategy is a feasible approach for the task allocation problem, and the resulted task assignment is near optimal. This means that the total cost of the team is minimized. A major advantage is its low computation cost

Accepted April 10, 2007

I. Introduction:

Recently military conflicts have demonstrated the strategic value of UAVs. The roles of UAVs are evolving from reconnaissance purpose to offensive mission as missile launching platform. The capabilities of UAVs will be further improved if multiple UAVs are cooperative [1]. Achievement of cooperation among UAVs requires a method of assigning tasks. TA is an important problem to minimize the overall team cost. Besides, the assigning algorithm must ensure that all targets are approached in an optimal manner.

There are two types of task allocation problems: static and dynamic. Static task allocation means that the assignment may be made at time t such that all of the UAVs are committed. while dynamic task allocation is made at any of several discrete points of time [2]. Determining which of U agents are assigned to which of T targets is a problem of order T^U in complexity, so task allocation is an NP complete optimization problem [3].

There are a wide variety of approaches that have been reported for solving the task allocation problem in various applications. They can be classified into the following categories: network flow optimization [4], market based approach [5], integer linear programming, [6] fuzzy approach [7], and genetic algorithms [8]. Because of the intractable nature of the TA problem and its importance in cooperative control, it is desirable to explore other avenues for developing good heuristicalgorithm for the problem. The genetic algorithm (GA) is an intelligent probaplistic search algorithm that models the process of nature selection and genetics [9]. It is an iterative algorithm that maintains a pool of feasible solutions for each iteration. The GA starts with a set of randomly selected chromosomes as the initial population that encodes a set of possible solutions. Variables of a problem are represented as genes in a chromosome, and chromosomes are evaluated according to their fitness values, which are obtained by evaluating the considered fitness

Recombination typically involves two operators: (1) crossover and (2) mutation. Genetic operators alter the composition of genes to create new chromosomes referred to as offspring. The selection operator is an artificial version of nature selection, a Darwinian survival of the fittest among populations, to create populations from generation to generation. Chromosomes with better fitness have higher probabilities of being selected in the next generation. After several generations, GA can converge to the best solution. GA has many advantages over other heuristic techniques. For example it can be implemented in a few lines of computer code, it requires only primitive mathematical operators, and it has high probability to escape local minima.

In this paper, an explanation of a genetic algorithm in dynamic case is introduced. For a UAV in a fleet to be autonomous, it has to compute its trajectory and specify its target in real time. If a pop-up threat is arised or if a member of the fleet is lost re-planning for the trajectory and reallocation for targets has to be done in real time, to optimize the overall mission cost i.e. to minimize the UAVs trajectories threats and length. Consequently, the fuel consumption and the Vehicles flying time are minimized. The proposed algorithm is implemented in dynamic situations. It gives the near optimal solutions in a few seconds which is suitable for fast reaction of the vehicles to the new situations. The allocations in this case may not be the same allocations produced when the mission started. The rest of this paper is organized as follows :

section 2 description of the GA algorithm, section 3 introducing GA algorithm for task allocations section 4 implementing GA for Dynamic allocations, section 5 the simulation results.

2. Description of the Genetic Algorithm:

The genetic algorithm is a stochastic optimization algorithm that was originally motivated by the mechanisms of natural selection and evolutionary genetics. Over the last decade, GA has been extensively used as search and optimization tools in various problem domains, including : science, commerce and engineering. The primary reasons for their success are their broad applicability, ease of use and global perspective. There are some differences between the GA and traditional searching algorithms. They can be summarized as follows [10]:

- The algorithm works with a population of strings, searching many peaks in parallel, as opposed to a single point.
- The GA works directly with strings of characters representing the parameter sets, not the parameters themselves.
- The GA uses probabilistic rules instead of deterministic rules.
- The GA uses objective function information instead of derivatives or other auxiliary knowledge.

GA is inherently parallel, because it simultaneously evaluates many points in the parameter space (search space). So, the GA has a reduced chance of converging to local optimum and would be more likely to converge to global optimum. It requires only information concerning the quality of the solution produced by each parameter set (objective function values). This differs from many optimization methods which require derivative information or, worse yet, a complete knowledge of the problem structure and parameters. Since the GA does not require such problem specific information, it is more flexible than that most search methods [2]. Typically, the GA is characterized by the following components:

- A genetic representation (or an encoding) for the feasible solution to the optimization problem.
- A population of encoded solution.
- A fitness function that evaluates the optimality of each solution.
- Genetic operators that generate a new population from the existing population.
- Control parameters.

The basic flow chart of the GA is illustrated in Fig. 1 where ($\varepsilon > 0$) a small number to check convergence.

3. The Proposed Technique:

3.1 Task Allocation (static situation):

For task allocation problems, applying the normal mutation and cross over procedures on a binary representation will lead to illegitimate solutions i.e. assigning a non existing targets to a UAV (it will produce targets number

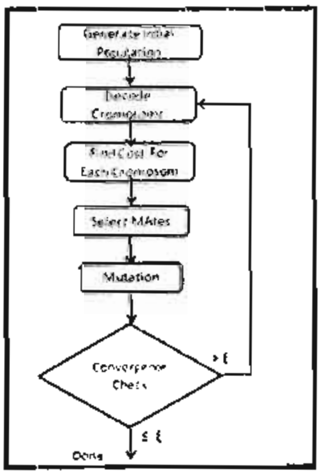

Fig. 1 The Procedure Of The Genetic Algorithm

more than the specified number of targets T). A number of researchers [9],[11],[12] have created operators that overcome this problem by implementing an operator called ordered crossover to be used. For targets allocation problems the same operator can be used, and the coding can be alphabetic or numeric. This operator builds offspring by choosing a subsequence of UAVs within a list of one parent. It also preserves the relative ordering of UAVs from the other parent.

Assuming the number of UAVs equals the number of targets i.e. U = T, for simplicity numeric codes for targets are used i.e. 1 2 3 4 5 6 7 8 9, and letters are used for UAVs: A B C D E F G H I. As an initial population the first parent P1 and the second parent P2 can be randomly generated as:

T 1 2 3 4 5 6 7 8 9
P1 B C H G E A D F I (1)
P2 F H C D G A I B E

This means that for P1 UAV B is assigned to target #1, C is assigned to target # 2, H is assigned to target # 3 and so on. Two children C1 and C2 are produced in the following way. First, two cut lines are chosen for C1 and C2 for the high fitness chromosomes. The segments between cut points are copied into the following offspring:

T 1 2 3 4 5 6 7 8 9
C1 X X X G E A D X X (2)
C2 X X X D G A I X X

Next, starting from the second cut point of one parent, the UAVs of the other parent are copied in the same order, omitting UAVs already present between the two cut lines.

T 1 2 3 4 5 6 7 8 9
C1 B F H G E A D I B (3)
C2 F B C D G A I E F

The position of the segments in the two children C1, C2 is the same and depends on the heights fitness. For one parent C1 a mutation operator can be done by exchanging places of randomly selected UAVs in the children chromosomes. Evaluation can be done by using a suitable fitness functions f. The fitness function for any chromosome i can be evaluated as $f_i = 1/cost$

or $f_i = Q / \cos(t_i)$, (4)

where Q is a suitable weighting factor. As the cost of the chromosome is reduced its fitness is increased. Consequently, its probability to be repeated in the next offspring is increased. It is called the repetition rate (RR) in the next offspring. This RR is evaluated as the fitness of the individual chromosome over the average fitness i.e.

$RR_i = f_i / (\sum (f_i) /n)$ (5)

The chromosomes that have values (≥ 1) are optimal and kept in the next offspring. So, the solution is approaching optima by increasing all RRi to be ≥ 1 ($i = 1,\ldots,n$) with iterations. It must be noted that if the RR increased than 1 it is considered to be only 1 since each UAV is assigned to one target only.

The optimal solution for the team is reached when RRi ≥ 1 for $i=1,\ldots,n$ i.e.

$$\text{average RR} = \sum_{i=1}^{n}(RRi/n) \geq 1 \qquad (6)$$

Given the cost functions J for n UAVs to m targets, the algorithm can be explained as follows:

<u>Step 1</u> initiate two random population P1, P2 as shown in Eqn.1

<u>Step 2</u> evaluate the cost J_i for every chromosome in the population. Then compute the fitness function f_i by Eqn.4.

<u>Step 3</u> compute the repetition rate RR for every chromosome by Eqn. 5 for P1, P2.

<u>Step 4</u> for any chromosome in P1 where RR$\geq$1 determine two cut lines around it. The same two cut lines for P1 are determined for P2. If the chromosome that have RR $\geq$1 are not adjacent then more than two lines must be determined.

<u>Step 5</u> in P1 keep all chromosome in between the two lines in the next offspring C1 as shown in Eqn.2. Execute cross over starting from the first cut line in P2. Omit any existent chromosome (already kept in C1) as shown in Eqn.2.

<u>Step 6</u> execute the mutation process, simply by exchanging the position of two chromosomes in C1.

<u>Step 7</u> repeat the same steps 4, 5, 6 for P2. till fulfilling the exit condition.

<u>Step 8</u> repeat step 3 to step8 till optimal or near optimal solution is reached. i.e. the converge < ε The solution can be detected from the solution average fitness $\sum (f_i) /n$

3.2. Dynamic Reallocation:

The autonomous UAVs plan their trajectories, and the GA algorithm assigns them to targets.

These processes are in real time, while the vehicles are executing the planned trajectories. The situation in the battle field can be changed due to any unexpected reasons, e.g. new threats are detected, new targets are explored, a team member is lost, ...etc. In this case, the team re-plans trajectories according to the new situation. The GA is used to reallocate the team members.

The initial offspring is not randomly chosen, but the last allocation before the new situation airside is the beginning population. So, if at time (t) a new situation airside and the allocation is C1(t), C2(t), starting from the current positions of the (n) vehicles the real time trajectory planning algorithm computes the new feasible trajectories to the m targets. The (m x n) minimum cost functions J of these trajectories are produced and passed to the TA algorithm. By its turn the TA reallocates the vehicles from its current positions to the targets. The same algorithm explained in 3.1 is used considering the initial population as: P1 = C1(t), and P2 = C2(t)

4. Simulation Results :

In a real mission, the number of vehicles (m) may be more or less than the number of specified targets (n). So, some targets may be assigned to more than one vehicle (in the first case where m > n). In the other case (m < n) a vehicle may be assigned to more than one target one after another. In this paper, the following assumptions are considered:

- The number of vehicles and targets are equal i.e. m = n, and each target is assigned to one vehicle.
- The vehicles are equipped with the necessary systems such as: sharing information transceivers, anti-jammers, anti-decoyers, anti-collation, and different flying phases algorithms (eg. taking off -landing- attacking) .. etc.

4.1 Static Allocation:

The algorithm is applied to allocate 6 UAVs (A to F) to 6 targets (1 to 6), the costs of trajectories are shown in the table 1.

These costs are produced from a trajectory planning algorithm explained in Fig. 2. The cost function J for each trajectory is composed of two elements.

$$J = J_l + J_{th} \quad (7)$$

Where J_l is the length cost and J_{th} is the threat cost

Table 1 The Cost Functions For 6 UAVs Trajectories To 6 Targets

T \ U	A	B	C	D	E	F
1	1215	976	1170	2190	2054	2223
2	1214	1023	1237	1420	1972	1672
3	1557	1397	1617	1498	1526	1280
4	1456	1241	1458	1553	1684	1418
5	1379	1225	1424	1759	1860	1597
6	1916	2354	1498	2036	1328	1291

The task assignment is obtained in 5 iterations as shown in Table2, with the maximum total fitness 5.661, and average RR ratio = (5.661/6) = 0.94 which is sufficient for a near optimal solution (optimal solution is achieved by verifying Eqn. 6.) (if RRi>1 it is considered only 1, since every UAV is assigned only to one target). Fig.3 explains these trajectories for vehicles over a hostile territories. The trajectories are planned to avoid the surface to air missiles (SAMs) sites and to minimize the path length and probabilities of detection by the radar sites. The GA is used to allocate each vehicle to a target. The main objective is to optimize the overall cost for the team as a whole.

4.2 Dynamic Allocation:

When the team members are executing their trajectories to the assigned targets a sudden threat is detected (Shown as the shaded area in Fig. 3). The threat is detected when the vehicles are on the positions shown in Fig.3 assuming the team has a rendezvous arrival time.

The trajectory planning algorithm re-plans a new group of the minimum cost feasible trajectories and computes their costs. The costs are listed in Table 3. The GA is used on line to reallocate the targets as shown in Table 4.

Table 2 The Fitness Functions For Each Generation

	Generation 1		Generation 2		Generation 3		Generation 4		Generation 5	
T	P_1	$F_i/(\Sigma(f_i)/n)$	P_2	$F_i/(\Sigma(f_i)/n)$	P_3	$Fi/(\Sigma(f3_i)/n)$	P_4	$F_i/(\Sigma(f4_i)/n)$	P_5	$F_i/(\Sigma(f_{5i})/n)$
1	B	1.44	B	1.34	B	1.43	B	1.34	B	1.29
2	C	1.137	C	1.06	C	1.132	C	1.06	C	1.03
3	D	0.939	D	0.877	E	0.917	D	0.877	F	0.98
4	F	0.99	A	0.902	F	0.987	A	0.902	D	0.81
5	E	0.756	F	0.822	D	0.796	F	0.822	A	0.98
6	A	0.73	E	0.989	A	0.73	E	0.989	E	0.953
Σfitness	5.415			5.59		5.43		5.59		5.661
Σcost	8905			8092		8832		8092		7753
Computation Time *										0.49 second

Fig.3 shows the new trajectories after the new allocation. the population before the popup threat aroused (generation 5 in Table 2) is used as an initial population. The algorithm reached a near optimal solution (average RR= (5.976/6) = 0.996. in 0.36 seconds.

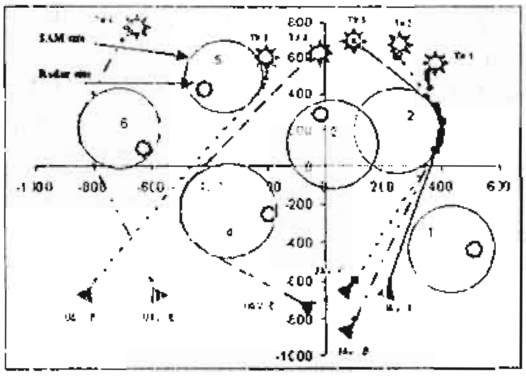

Fig. 2 Targets Allocations For 6 Vehicles

Table 3 The Costs OF New Trajectories After a Pop-up Threat Is Detected

T \ U	A	B	C	D	E	F
1	583	468	562	1051	986	1067
2	583	491	594	682	947	803
3	747	671	776	719	732	614
4	699	596	700	1603	808	681
5	662	588	684	844	893	767
6	920	1130	719	977	637	620

Table 4 Reallocation Of Targets

T	P	$Fi/\Sigma(Fi/n)$
1	B	1.30
2	D	0.90
3	F	1.00
4	C	0.88
5	A	0.93
6	E	0.966
Σf_i		5.976
Σ cost		1802
Computation Time		0.36 second

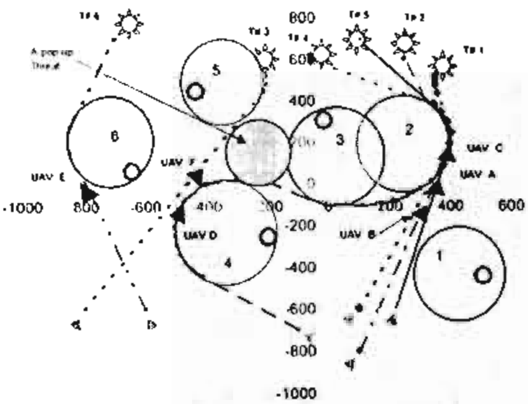

Fig.3 The New Trajectories For Newly Assigned Targets

* On P IV 3 GHz, 256 MB Memory, 800 MHZ

Major advantage of GA algorithm is that it searches for optimization (as stated in Eqn. 6) in a parallel manner. Consequently, it produces the near optimal solution in a few seconds. This is a vital element for the vehicle autonomy i.e. to plan its trajectories to its assigned targets in real time. It is worth noting that, the trajectory of the vehicle D to its pre-assigned target (target 4) is no longer feasible because of the new threat (Fig.3). Hence, its new trajectory is forced to turn around the new threat along with two additional SAM sites namely number 2 and 3. The cost of that trajectory is increased as explained in table 3. From table 4 and Fig.3 it is clear that the UAVs C and D switched their targets to optimize the total cost of the fleet. The trajectories and allocations of the other vehicles A,B,E, and F are the same as before the re-planning. The average fitness of the group is 996.

The main results of the algorithm are (1) the search for optimal allocation is inherently parallel and very fast (the computation time is stated in table 2 and table 4). (2) the resultant allocation can be optimal or near optimal (average RRi ≈1) (3) the algorithm can fall on local minima so the algorithm has to observe if the solution failed in a local minima or not. However escaping from it is easy and can be done by exchanging two vehicle symbols positions. (4) the result for different initial population can be different, especially if the member of UAVs is limited. (5) GA algorithm enables the autonomy of the UAVs

Conclusions:

This paper introduces a genetic algorithm for assignment of autonomous multi unmanned air vehicles. The assignments are produced in both static and dynamic environments. The application This a crucial issue for vehicles safety, fuel consumption and flying time. b) The optimization process is inherently parallel, so the computation cost is very small. This permits the UAVs autonomy and adaptively to the dynamic situations by re-assigning the vehicles according to the new costs in real time. c) The algorithm can simply detect and escape local minima. d) The algorithm is simply coded. e) an important advantage of the algorithm is that the computation cost dos not increase polynomially with the number of vehicles and the number of targets. Consequently, the allocation problem is no longer NP-complete.

References

1. David Bookstber "Unmanned Combat Aerial Vehicles", "Flight International ", 1998.
2. Jose B., Cruez J. "Genetic Algorithm for Task allocation in UAV Cooperative Control, "Genshe C., AIAA Guidance, Navigation, and Control Conference, August 2003, Austin, Texas.
3. Paul Juell, Amal S. Perrera, Kendall E. Nygdrd "Genetic Algorithm to Improve a solution for a General Assignment Problem", North Dakotta State University 2002.
4. Ibarraki T. and Katoh N., "Resource Allocation Problem: algorithmic approaches", Cambridge, Mass MIT Press. 1988.
5. Chandler, P.R., Patcher, M., Nygard, K.E. and Swaroop. D. "Cooperative Control for Target Classification". Cooperative Control and Optimization (R. Murphey and P. M. Pardalos editor), Kluwer Acedemic Publishers, 2002.
6. Tierno, J.E., "Distributed autonomous control of concurrent combat tasks," Proceeding of the American Control Conference, Arlington, VA, June 25-27.2001, PP37-42.
7. Garagic, D. and Cruz, J.B., "Target Allocation Using a Binary Integer Programming with Fuzzy Objective, "OSU Workshop for DARPA MICA Program, November 08, 2002.

shows a lot of advantages of the algorithm in static and dynamic targets allocation such as:
a) The produced solution is optimal or near optimal i.e. minimizing overall the fleet trajectories cost.

8. Griggs B.J., Parnell G.S., and Lehmkuhl, "An Air Mission Planning Algorithm Using Decision Analysis and Mixed Integer Programming" Operations Research , 45(5), 1997, PP662-676.

9 George F. Luger, "Artificial Intelligence Structures and Strategies for Complex Problem Solving" Person Education (Singapore) Pre. Led, forth edition., 2002.

10. Armingol J.M., Moreno L., Escalera A.d., and Salichs M.A., "A Genetic Algorithm For Mobil Robot Localization Using Ultrasonic Sensors". Journal of Intelligent and Robotic Systems, Vol. 34, N.2, PP 135-154, 2002

11. Davis L. "Applying adaptive algorithm to Epistatic Domains, Proceedings of the International Joint Conference on Articial Intelligence", PP-162-164, 1985.

12. Hill Scale, N.J. Eribam, Oliver, I.M., Smith, D J. and Holland, J.R.C."A study of permutation crossover operator's on the Traveling Salesman problem". Proceedings of the second international conference on Genetic Algorithms, PP 724-240, 1987.