THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 13, Number 4/2012, pp. 295–301

A COMBINATION OF SWEEP ALGORITHM AND ELITE ANT COLONY OPTIMIZATION FOR SOLVING THE MULTIPLE TRAVELING SALESMAN PROBLEM

Majid YOUSEFIKHOSHBAKHT[1], Mohammad SEDIGHPOUR [2]

[1] Islamic Azad University, Department of Mathematics, Hamedan Branch, Hamedan, Iran
Email: khoshbakht@iauh.ac.ir
[2] Islamic Azad University, Department of Mathematics, Hamedan Branch, Hamedan, Iran
Email: sedighpour@iauh.ac.ir

The aim of this paper is to introduce a hybrid two-phase algorithm called SW+ AS_{elite} for solving the MTSP which can be explained as the problem of designing collection of routes from one depot to a number of customers subject to side constraints. At the first stage, the MTSP is solved by the sweep algorithm, and at the second stage, the Elite Ant Colony Optimization (AS_{elite}) and 3-opt local search are used for improving solutions. The performance of our algorithm has been tested on 6 MTSP benchmark problems and it shows that the proposed SW+ AS_{elite} performs well and is quite competitive with other meta-heuristic algorithms.

Key words: Elite Ant colony optimization, multiple traveling Salesman problem, sweep algorithm, NP-hard problems.

1. INTRODICTION

A lot of research has been carried out in the field of logistics from the traveling salesman problem to complex dynamic routing problems. Among the prominent problems in the distribution and logistics are the Traveling Salesman Problem and its extensions. These problems have been widely studied for many years, mainly because of their applications in real world logistics and transportation problems.

The multiple traveling salesman problem is a generalization of the well-known traveling salesman problem [3], where one or more salesman can be used in the solution. The MTSP can in general be defined as follows.

Given a set of nodes, let there are m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are called intermediate nodes. Then, the MTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. An example is depicted in Fig.1, where $m = 3$ and $n = 10$.

In the following parts of this paper, in Section 2, the related work will be described. In section 3, we will explain the formulation of MTSP. Then, at the beginning of section 4, we specially explain the sweep algorithm and Elite Ant Colony Optimization and then the combination of AS_{elite} and sweep algorithms are extendedly explained and its efficiency and performance will be described. In section 5, the proposed algorithm will be compared with some of the other algorithms on standard problems, which are included in the MTSP library, and finally in section 6, the conclusions are presented.

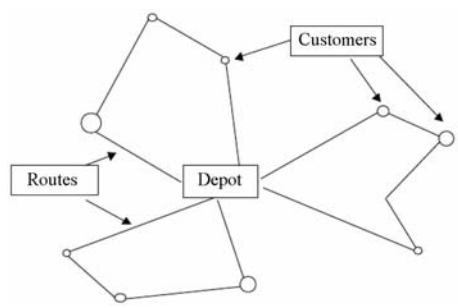

Fig 1 – Sample of solving MTSP.

2. RELATED WORK

For many years, researchers have paid so much attention on the traveling salesman problem while there are many problems expanded by it, like the vehicle routing problem and the multiple traveling salesmen problem and so on [14]. Besides, it has many applications in other problems including the Print press scheduling [8], Crew scheduling [18], School bus routing problem [2], Interview scheduling [7], Mission planning [4], Hot rolling scheduling [20], Design of global navigation satellite system surveying Networks [17], etc.

There are a broad variety of MTSPs and a wide literature on this class of problems. The variants include MTSP with pickup and delivery, time windows, multi-depot MTSP and others. There have been important advances in the development of exact and approximate algorithms for solving the MTSP which is one of the most important combinatorial optimization problems. There are several exact algorithms of the MTSP such as Cutting Planes algorithm [10], Branch-and-Bound algorithm [1], and Lagrangian algorithm [22].

Because of the fact that TSP belongs to the class of NP-hard problems [15], it is obvious that MTSP is an NP-hard problem. This means that the MTSP solution time grows exponentially with the increase in distribution points thus exact algorithms are not capable of solving problems for large dimensions. On the other hand, heuristics are thought to be more efficient for complex MTSP and have become very popular for researchers. A lot of algorithms have been performed on the MTSP including heuristic approaches such as *k*-opt approach [13], minimum spanning tree [11], and self-organizing NN approach [21].

Although heuristic methods solve NP-hard problems, they become trapped in local optima and cannot gain a good suboptimal solution. As a result, in the last 30 years, a new kind of approximate algorithm called meta-heuristics has emerged which basically tries to combine basic heuristic methods into higher level frameworks aimed at efficiently and effectively exploring a search space. Some of the well known meta-heuristics which have more ability for finding a optimal solution are the genetic algorithm [5], the neural networks [12], the tabu search [16] and the ant colony optimization [23, 6].

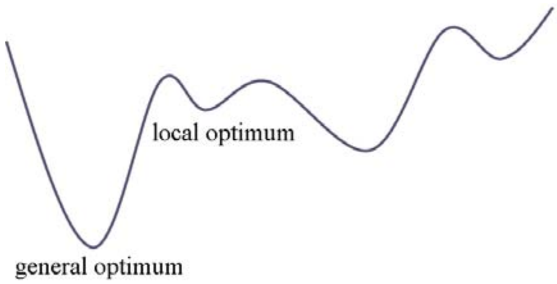

Fig 2 – Local and general optimums.

3. MATHEMATICAL MODEL

The MTSP can be explained mathematically as follows.

Let *G* (*V*, *A*) be a perfect undirected connected graph with a vertex set $V = \{0,1,...,n\}$ and an edge set $A = \{(i, j) : i, j \in V, i \neq j\}$ .If the graph is not perfect, we can replace the lack of any edge with an edge that has an infinite size. Vertex 0 represents the depot, each other vertex $i \in V - \{0\}$ is a customer, and $c_{ij}\ (c_{ij} = 0,\ \forall i = j)$ represents the distance from node *i* to node *j*. By introducing variables x_{ij} to represent the tour of the salesman travels from city *i* to city *j*, one of the common integer programming formulations for the MTSP can be written as:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}, \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 2,...,n, \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = m, \qquad j = 1, \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 2,...,n , \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = m, \qquad i = 1 \tag{5}$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 \ (\phi \neq S \subset \{2,...,n\}, \quad |S| \geq 2, \tag{6}$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 \ (\phi \neq S \subset \{2,...,n\}, \quad |S| \geq 2, \tag{7}$$

$$x_{ij} \in \{0,1\}. \tag{8}$$

The objective function (1) is simply to minimize the total distance traveled in a tour. Constraint sets (2) and (3) ensure that the salesman arrives once at each node and m times at the depot. Constraint sets (4) and (5) ensure that the salesman leaves each node once and leaves the depot m times. Constraint sets (6) and (7) are to avoid the presence of sub-tours for each salesman. Finally, Constraint set (8) defines binary conditions on the variables.

4. OUR APPROACH

In this section, first the Sweep Algorithm is presented and then the Elite Ant Colony System will be explained and, finally, the proposed algorithm will be analyzed in more detail.

4.1. Sweep Algorithm

Gillett and Miller proposed a sweep algorithm for Euclidean networks, which ranks and links demand points by their polar coordinate angle. The polar coordinate angle is calculated as follows:

$$An(i) = \arctan\{(y(i) - y(0))/(x(i) - x(0))\}. \tag{9}$$

This results in $-\pi < An(i) < 0$ if $y(i) - y(0) < 0$ and $0 \leq An(i) \leq \pi$ if $y(i) - y(0) \geq 0$. In this algorithm, which is one of the famous and powerful heuristic algorithms, the polar coordinates of all customers are calculated, where the center is the depot and an arbitrary customer is chosen to be at an angle 0.

$$0 = An(1) \leq An(2) \leq ... \leq An(n). \tag{10}$$

Then, sweeping (clockwise or counterclockwise) is started from customer i, which has not been visited, from the smallest angle to the largest angle until all customers are included in a tour.

4.2. Elite Ant Colony System

The first algorithm of ACO family was the Ant System algorithm (AS) used for solving small scale TSP instances. In the face of gaining good solutions for small scale problems of TSP type, the AS algorithm failed to reach an acceptable efficiency in large scale problems in comparison with the famous algorithms of that time. Much effort was performed to solve the problem; the first modification applied on the AS algorithm was the usage of elitist strategy, published by Dorigo et al in 1996 [16]. Based on this algorithm, in addition to the local releasing of pheromone on the arcs which the ants have passed through, the arcs belonging to the best route (T^*) are released with pheromone and are encouraged with the constant coefficient e in the following way;

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} e / L^{gb}(t) & (i,j) \in T^* \\ 0 & (i,j) \notin T^*, \end{cases} \tag{11}$$

This process causes that the arcs belonging to the best route in any iteration are more highlighted, and to be updated according to the value of the best route L^{gb}. Note that, the above operator indicates; the less the value of L^{gb}, the more pheromone released on the arcs.

In spite of better results obtained by AS_{elite} algorithm, premature convergence occurred during the calculation process. In other words, releasing global pheromone with constant coefficient leads to a very fast and early concentration of searching procedure around suboptimal solutions. In addition, this will cause an early stagnation of premature search and settling for a local optimal. Premature convergence makes ants trail similar route and gains similar solutions for several times. In this way we cannot get to better solutions. It shows that system stops from finding possibly better routes and cannot gain a better tour.

Updating the pheromone simulates the changes in values of pheromone in any iteration and mainly it is one of the reasons that algorithms are different. Generally, two operations motivate this updating procedure in AS_{elite} algorithm:

- 1st, releasing new pheromone on the arcs, locally and globally; this operation leads to increase pheromone on the arcs.
- 2nd, evaporation of pheromone; this operation leads to decrease pheromone on the arcs with constant rate ρ, in other words, at end of each iteration of algorithm, the value of pheromone left on the arcs is decreased by the constant coefficient ρ. Thus, the new footprint of pheromone has an average weight between the value of the pheromone left on the arcs, and the value of new pheromone released in the arcs.

Thus the formula of updating pheromone in the AS_{elite} algorithm is:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) + \Delta\tau_{ij}^{gb}(t), \qquad (12)$$

where ρ is the evaporation rate, which is a constant value in [0,1] domain. It regulates the decrease of phermone on arcs

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & (i,j)\in T^{k} \\ 0 & (i,j)\in T^{k}, \end{cases} \qquad (13)$$

T^{k} – the collection of arcs passed over by the ant *k*.

$\Delta\tau_{ij}^{k}(t)$ – as mentioned above, is the formula of updating the pheromone. Thus ants passing over the arc between nodes i and j, release some pheromone on it. The value of released pheromone is one over the value passed yet.

$\Delta\tau_{ij}^{gb}(t)$ – pointed above.

4.3. Proposed Algorithm

From another viewpoint, there are also two heuristic groups for solving combinatorial optimization problems: construction algorithms that produce a feasible solution themselves, and improvement algorithms that can improve the solutions although they cannot generate a feasible solution. Here, a construction algorithm, namely sweep algorithm, and two improvement algorithms, namely AS_{elite} and 3-opt local search, are used. Furthermore, the AS_{elite} is used for improving every route of the salesman; however, the nodes of each salesman should be unchanged. On the other hand, 3-opt local search is applied for changing nodes and improving each salesman. In other words, if this algorithm is separated into two parts, the first one to improve the route of the salesman without changing the salesman's nodes and the second one to improve it only by changing the salesman's nodes, then the AS_{elite} does operation 1 and 3-opt local search does operation 2.

In this method, first the nodes that should be visited by salesmen are ordered with respect to the depot by sweep algorithm, and then they are set in the array. Second, all of the states in which the above-mentioned order is preserved in spite of a change in their locations are obtained. For example, if 1 2 3 is the order obtained for a feasible solution, then orders 2 3 1 and 3 1 2 are set in other arrays. Then for each array, the salesman starts to move from the depot and visit the nodes in the arrays in the order described until it is not possible to add a further node because of n/m in table1. This means that if the number of customer visited by salesman is more than n/m, it returns to the depot and repeats these steps until there is not any node to be visited. When salesman routes are obtained the AS_{elite} is implemented for every route until the best route is obtained. An example is depicted in Fig. 3, where $n = 11$, $m = 3$ and $n/m \simeq 3.7$.

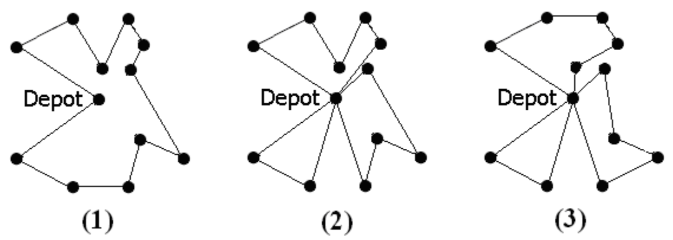

Fig. 3 – Sweep Algorithm (1), (2) and implementing AS_{elite} regarding the salesman's constraints (3).

Furthermore, the vast literature on meta-heuristics tells us that a promising approach to obtaining high-quality solutions is to couple a local search algorithm with a mechanism to generate initial solutions. ACO's definition includes the possibility of using local search as daemon actions. Daemon actions are used to implement centralized actions which cannot be performed by a single ant. An example of daemon actions is the activation of a local optimization procedure. In implementation, 3-opt local search procedure is used to obtain more improvement in the algorithm's performance.

This algorithm is shown below, which is based on omitting three arcs of the tour that are not adjacent and connecting them again by another method. It can be noted that there are several routes for connecting nodes and producing the tour again, but a state that satisfies the problem's constraints is acceptable. So, this unique tour will be accepted only if, first, the above constraints and u in table 1 are not violated and, second, the new tour produces a better value for the problem than the previous solution. It should also be noted that omitting three arcs and again connecting them are started when a better solution is found. When all of these n arrays are gained, the best one is selected as the best solution and the algorithm will be finished.

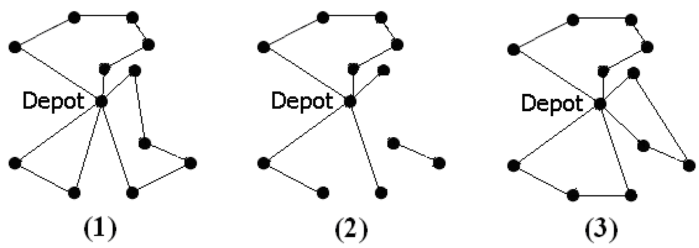

Fig. 4 – 3-opt local search.

So, it must be noted that the strength of this algorithm in comparison with other algorithms such as ant colony optimization or tabu search is that although the hybrid algorithm is composed of two heuristic and one meta-heuristic methods, first, it takes less time to obtain the solution and, second, the algorithm does not need many iterations for gaining a better solution. On the other hands, when the AS_{elite} is activated, the number of nodes is decreased and the algorithm can reach a good solution during the execution. In addition, because of the lack this division in other meta-heuristic algorithms, the size of the problem is large and solutions obtained are usually less stable compared to those of this algorithm in various executions.

5. COMPUTATIONAL EXPERIMENTS

In this section, some numerical results of comparison between the proposed algorithm and some meta-heuristic algorithms are presented. These algorithms are applied and tested on several instances from TSP problems of TSPLIB including Pr76, Pr152, Pr226, Pr299, Pr439 and Pr1002. For each instance, the number of customers, n, the number of salesman, m, and the max number of customers that a salesman can visit, u, is presented.

The algorithms are coded by C++ language and implemented on a Pentium 4 PC at 3GHZ (512MB RAM). Furthermore, the pack of optional parameters obtained through several tests is as follows:

$$\alpha = 1, \beta = 5, \rho = 0.5, \ Q = 100, \ e = 1 \tag{14}$$

In these tests we compared the efficiency and performance of the proposed algorithm (PA) with modified genetic Algorithm (MGA) [19] and Modified Ant Colony Algorithm (MACA) [9]. The results of this simulation are given in Table 1 and Table 2. In Table 1, for each algorithm the average, AS, and the best of obtained solutions, BS, are reported. The results of this comparison show that the proposed algorithm gains worse solutions than the MGA in Pr76, which are not large scale problems, and it gains better solutions than the MGA in following scale problems such as Pr152, Pr226, Pr229, Pr439 and Pr1002. Furthermore, the results indicates that although the MACO gives a better solution that the proposed algorithm for Pr226, this algorithm cannot maintain this advantage in the proceeding examples and the proposed algorithm yields better solutions than the MACO for other instances including Pr76, Pr152, Pr299, Pr439 and Pr1002.

On the other hand, Summary results over 10 runs of proposed algorithm, MACO and MGA are also given in Table 1. The AS reported in this table represents the average obtained solutions per run over 5 MTSP problem instances. Results indicate that proposed algorithm has produced 5 new average solutions and these new best known solutions are highlighted in bold letters in Table 1.

Table 1

Comparison of algorithms for standard problems of MTSP

PA			MACO			MGA			u	m	n	Instance
BS	AS	T	BS	AS	T	BS	AS	T				
157495	**157562**	19	178597	180690	51	157444	160574	43	*20*	*5*	*76*	*Pr76*
127791	**128004**	41	130953	136341	128	127839	133337	91	*40*	*5*	*152*	*Pr152*
167665	**168156**	62	167646	170877	143	166827	178501	165	*50*	*5*	*226*	*Pr226*
81998	**82195**	65	82106	83845	288	82176	85796	363	*70*	*5*	*299*	*Pr299*
161725	**162657**	95	161955	165035	563	173839	183698	623	*100*	*5*	*439*	*Pr439*
379871	**381654**	186	382198	387205	2620	427269	459179	2892	*220*	*5*	*1002*	*Pr1002*

Table 2 shows the computation times with benchmark problem instances that have been reported by various algorithms. Computational experiment has shown that in general the proposed algorithm gives better results compared to the existing solution methods for MTSP in terms of the CPU time. Overall, ACS has produced 5 new best known solutions for MTSP problem instances available in the literature.

Table 2

Comparison of CPU times with published results

PA	MACO	MGA	Instance
19	51	43	*Pr76*
41	128	91	*Pr152*
62	143	165	*Pr226*
65	288	363	*Pr299*
95	563	623	*Pr439*
186	2620	2892	*Pr1002*

6. CONCLUSION

In this paper, a hybrid algorithm combining AS_{elite}, Sweep algorithm and 3-opt local search was proposed for solving the Multiple Traveling Salesman Problem. SW+ AS_{elite} is more efficient than modified ant colony optimization and modified genetic algorithm especially for problems. It seems that the combination of the proposed algorithm and genetic algorithm, tabu search and simulated annealing algorithms will yield better results. Besides, using this proposed algorithm for other versions of the multiple traveling salesman problem and also applying this method in other combinational optimization problems including the vehicle routing problem, School bus routing problem and the sequencing of jobs are suggested for future research.

REFERENCES

1. AI, A. and KENNINGTON, J. L., *Exact solution of multiple traveling salesman problems,* Discrete Applied Mathematics, **13**, pp. 259–276, 1986.
2. ANGEL, R. D., CAUDLE, W. L, NOONAN, R., WHINSTON, A., *Computer assisted school bus schedulin,* Management Science, **18**, pp. 279–288, 1972.
3. BEKTAS, T., *The multiple traveling salesman problem: an overview of formulations and solution procedures,* Omega, **34**, pp. 209–219, 2006.
4. BRUMMIT, B., STENTZ, A., GRAMMPS: *a generalized mission planner for multiple mobile robots*, Proceedings of the IEEE international conference on robotics and automation, May 1998.
5. CARTER, A. E., RAGSDALE, C. T., *A new approach to solving the multiple traveling salesperson problem using genetic algorithms,* European Journal of Operational Research, **175**, pp. 246–257, 2006.
6. GHAFURIAN, S., JAVADIAN, N., *An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems*, Applied Soft Computing, **11**, *1*, pp. 1256-1262, 2010.
7. GILBERT, K. C., HOFSTRA, R. B., *A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem,* Decision Sciences, **23**, pp. 250–259, 1992.
8. GORENSTEIN, S., *Printing press scheduling for multi-edition periodicals,* Management Science, **16,** *6*, pp. 373–383, 1970.
9. JUNJIE, P., DINGWEI, W., *An ant colony optimization algorithm for multiple traveling salesman Problem*, In ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control, pp. 210–213, Washington, DC, USA, 2006. IEEE Computer Society.
10. LAPORTE, G., NOBERT, Y. A., *cutting planes algorithm for the m-salesmen problem,* Journal of the Operational Research Society, **31**, pp. 1017–1023, 1980.
11. MALIK, W., RATHINAM, S., DARBHA, S., *An approximation algorithm for a symmetric Generalized Multiple Depot, Multiple Travelling Salesman Problem,* Operations Research Letters, **35**, *6*, pp. 747–753, 2007.
12. MODARES, A., SOMHOM, S., ENKAWA, T., *A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems*, International Transactions in Operational Research, **6**, *6*, pp. 591–606, 1999.
13. POTVIN, J., LAPALME, G., ROUSSEAU, J., *A generalized k-opt exchange procedure for the MTSP*, Information Systems and Operations Research, **27**, *4*, pp. 474–481, 1989.
14. RATHINAM, S., SENGUPTA, R., *Lower and upper bounds for a symmetric Multiple Depot, Multiple Travelling Salesman Problem*, Research Report, UCB-ITSRR- 2006-2, Institute of Transportation Studies, 2006.
15. RUSSELL, R. A., *An effective heuristic for the m-tour traveling salesman problem with some side conditions,* Operations Research, **25**, *3*, pp. 517–524, 1977.
16. RYAN, J. L., BAILEY, T. G., MOORE, J.T., CARLTON, W.B., *Reactive Tabu search in unmanned aerial reconnaissance simulations*, Proceedings of the 1998 winter simulation conference 1, 1998, pp. 873–879.
17. SALEH, H. A., CHELOUAH, R., The design of the global navigation satellite system surveying networks using genetic algorithms. Engineering Applications of Artificial Intelligence, **417**, pp. 111–122, 2004.
18. SVESTKA, J. A., HUCKFELDT V. E. *Computational experience with an m-salesman traveling salesman algorithm,* Management Science, **19,** *7*, pp. 790–799, 1973.
19. TANG, T., LIU, J., *multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex*, European Journal of Operational Research, **24**, pp. 267- 282, 2000.
20. TANG, L., LIU, J., RONG, A., YANG, Z., *A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex*, European Journal of Operational Research, **124**, pp. 267–82, 2000.
21. VAKHUTINSKY, I. A., GOLDEN, L. B., *Solving vehicle routing problems using elastic net,* Proceedings of the IEEE international conference on neural network, 1994, pp. 4535–4540.
22. YADLAPALLI, S., MALIK, W. A., DARBHA, S., PACHTER, M., *A Lagrangian-based algorithm for a Multiple Depot, Multiple Traveling Salesmen Problem*, Nonlinear Analysis: Real World Applications, **10**, *4*, pp. 1990-1999, 2009.
23. YOUSEFIKHOSHBAKHT, M., ZAFARI, A., *A new ant colony algorithm for solving multiple traveling salesman problem*, The 2th Joint Congress on Intelligent and Fuzzy Systems (ISFS2008), 28–30 October, 2008, Malek-Ashtar University of Technology, Tehran, Iran (in persion).

Received May 11, 2011