論文
3-2-7

Distributed Genetic Algorithm using aster/slave model for the TSP
(TSP를 위한 마스터/슬레이브 모델을 이용한 분산유전 알고리즘)

(Jung-Sook Kim)
김 정 숙*

ABSTRACT

As the TSP(Traveling Salesman Problem) belongs to the class of NP-complete problems, various techniques are required for finding optimum or near optimum solution to the TSP.

This paper designs a distributed genetic algorithm in order to reduce the execution time and obtain more near optimal using multi-slave model for the TSP. Especially, distributed genetic algorithms with multiple populations are difficult to configure because they are controlled by many parameters that affect their efficiency and accuracy. Among other things, one must decide the number and the size of the populations (demes), the rate of migration, the frequency of migrations, and the destination of the migrants. In this paper, I develop random dynamic migration rate that controls the size and the frequency of migrations. In addition to this, I design new migration policy that selects the destination of the migrants among the slaves

요 약

외판원 문제는 NP-완전 문제 중의 하나로, 외판원 문제에 대한 최적해를 구하거나 근사해를 구하는 다양한 방법들이 개발되고 있다.

본 논문에서는 마스터/슬래이브 모델을 이용하여 외판원 문제를 해결하는 효율적인 분산 유전 알고리즘을 개발하였다. 특히 다중 후보해를 가진 분산 유전 알고리즘을 수행할 때, 고려해야 할 가장 중요한 요소는 후보해들 간의 개체들을 어떤 노드의 후보해 개체와 교환할 것인가와 어떤 개체들을 선택해서, 얼마만큼의 개체를 이동시킬 것인가가 중요하게 고려되어야 한다. 따라서 본 논문에서는 교환해야 할 개체의 크기를 임의로 생성하여 동적으로 변경하면서 교환하는 방법을 개발하였고, 또한 개체들이 교환되어질 슬래이브들의 위치를 결정하는 이동 정책을 개발하고 실험하였다.

1. Introduction

In the traveling salesman problem, a set of N cities is given and the problem is to find the shortest route connecting them all, with no city visited twice and return to the city at which it started. In the symmetric TSP, *distance*(ci, cj) =

* 정회원 : 김포대학 컴퓨터계열 소프트웨어 개발 전공 전임강사

논문접수 : 2002. 1. 26.
심사완료 : 2002. 2. 16.

distance(cj, ci) holds for any two cities ci and cj , while in the asymmetric TSP this condition is not satisfied.

Since TSP is a well-known combinatorial optimization problem and belongs to the class of NP-complete problems, various techniques are required for finding optimum or near optimum solution to the TSP. The dynamic programming and the branch-and-bound algorithm could be employed to find an optimal solution for the TSP. In addition to the classical heuristics developed especially for the TSP, there are several problem-independent search algorithms which have been applied to the TSP for finding near optimum solution, such as genetic algorithm[1, 5, 8], ant colonies[2], neural networks[2] and simulated annealing[2, 6]. And parallel and distributed systems have emerged as a key enabling technology in modern computing. During the last few years, many distributed algorithms are designed in order to reduce the execution time for the TSP with exponential time complexity in the worst case[7].

Under the right circumstances, I propose a new distributed genetic algorithm with multiple populations using multi-slave model for the TSP. The distributed genetic algorithm is based on the distributed population structure that has the potential of providing better near optimal value and is suited for parallel implementation. And distributed genetic algorithm executes a conventional genetic algorithm on each of the distributed populations. But the distributed genetic algorithm with multiple populations is difficult to configure because they are controlled by many parameters that affect their efficiency and accuracy. Among other things, one must decide the number and the size of the populations, the rate of migration, the frequency of migrations, and the destination of the migrants. In this paper, I develop random dynamic migration window method that controls the size and the frequency of migrations. In addition to this, I design new genetic migration policy that selects the destination of the migrants among the slaves.

The rest of this paper is organized as follows. Section 2 presents the related works and section 3 explains random dynamic migration rate, and section 4 describes the distributed genetic algorithm using multi-slave. Section 5 summarizes the methods of the experiments and the result. The remaining section presents the conclusion and outlines areas for future research.

2. Related Works

2.1 Distributed Genetic Algorithms

Distributed genetic algorithms have received considerable attention because of their potential to reduce the execution time in complex application. The basic idea behind many parallel programs is to divide a task into chunks and solve the chunks simultaneously using multiple processors. One common method to parallelize genetic algorithm is to use multiple demes (populations) that occasionally exchange some individuals in a process called migration.

The first method to parallelize genetic algorithm is to do a global parallelization. A more sophisticated idea is used in coarse-grained parallel genetic algorithm. In this case the population of the genetic algorithm is divided into multiple subpopulations or demes that evolve isolated from each other most of the time, but exchange individuals occasionally. Sometimes coarse-grained parallel genetic algorithms are known as distributed genetic algorithms. The third approach in parallelizing genetic algorithm uses fine-grained parallelism. We can recognize that some very important issues are emerging. First, parallel genetic algorithms are very promising in terms of the gains in performance. Second, parallel genetic algorithms are more complex than their serial counterparts. In this paper, the distributed genetic algorithm is based on the multiple population structure that has the potential

of providing better near optimal value and is suited for parallel implementation. And distributed genetic algorithm executes a conventional genetic algorithm on each of the distributed populations

2.2 Genetic Algorithm for the TSP

The studies on genetic algorithms for the TSP provide rich experiences and a sound basis for combinatorial optimization problems. Major efforts have been made to achieve the following. 1. Give a proper representation to encode a tour. 2. Devise applicable genetic operators to keep building blocks and avoid illegality. 3. Prevent premature convergence. Permutation representation has an appeal not only for the TSP but also for other combinatorial optimization problems. This representation may be the most natural representation of a TSP tour, where cities are listed in the order in which they are visited. The search space for this representation is the set of permutations of the cities. For example, a tour of a 17-city TSP 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 is simply represented as follows: <1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17>. The strength of genetic algorithms arises from the structured information exchange of crossover combination of highly fit individuals. Until recently, three crossovers were defined for the path representation, PMX(Partially-Mapped), OX(order), and CX(cycle) crossovers. OX proposed by Davis builds the offspring by choosing a subsequence of a tour one parent and preserving the relative order of cities from the other parent. During the past decade, several mutation operators have been proposed for permutation representation, such as inversion, insertion, displacement, and reciprocal exchange mutation. Reciprocal exchange mutation selects two positions at random and swaps the cities on these positions. The reproduction simply copies the selected parents of all genes without changing the chromosome into the next generation.

Selection provides the driving force in a genetic algorithm, and the selection pressure is critical in it. In rank selection, the individuals in the population are ordered by fitness and copies assigned in such a way that the best individual receives a predetermined multiple of the number of copies than the worst one. TSP has an extremely natural evaluation function: for any potential solution (a permutation of cities), we can refer to the table with distances between all cities and we get the total length of the tour after n-1 addition operations.

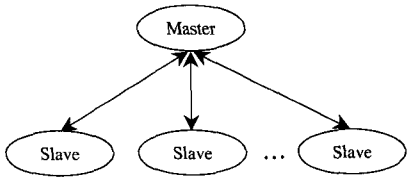
3. Random Dynamic Migration Rate

A migration rate that controls how many individuals migrate, and a migration interval that affects how often migrations occur. In the majority of coarse-grained parallel genetic algorithms, migration is synchronous meaning that it occurs at predetermined constant intervals, but it can also be asynchronous so that there is communication between demes only after some events occur.

Whenever the migration occurs, dynamic migration rate varies at random. The variation is from 1 to θ. The θ value is generated at random within 20% of the population size each migration. For example, first migration rate is 3 and second is 5, etc.

4. Distributed Genetic Algorithm using Multi-Slave

A new distributed genetic algorithm with multiple populations using multi-slave model for the TSP is illustrated in [Fig. 1].



[Fig. 1] Distributed genetic algorithm structure

In the multi-slave model, master maintains the lists of partial results that they are sent from slaves. At first, master generates the population and divides them into subpopulations. It sends them to slaves. The slaves execute a conventional genetic algorithm on their subpopulation: fitness evaluation, selection, crossover, mutation and periodically return their best partial results to the master. The master stores the partial results in lists. Then, master searches the lists and selects two slaves that have bad partial results and sends the migration window size to selected slaves. The selected slaves exchange the subpopulations according to the migration window methods. After the first migration occurs, slaves execute the conventional genetic algorithm on their fraction of the population and then send a result to master again. To select the destination of migrants again, master compares the lists and picks up θ slaves that have both bad partial result and less difference than threshold. I compute the difference as follows :

$$| (i - 1)_{th} \text{ partial result} - i_{th} \text{ partial result} | < \text{threshold.} \quad (1)$$

Threshold is variable. As the number of generation increase, threshold is decreasing. The master generates neighbors according to all possible permutations of selected slaves and evaluates all neighbors and then selects the best one as destination of the migrants to avoid the premature convergence. Master sends the migration window size to selected slaves. The selected slaves again exchange the subpopulations according to the migration window methods. Continues these works to find near optimal for the TSP.

The migration events occur per the given migration frequency rather than every generation to reduce the communication time.

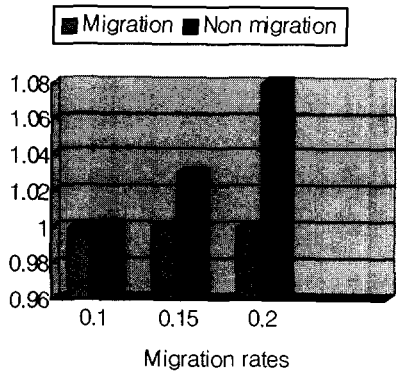
5. Experiments and Results

The hardware used is a collection of the PCs connected with a 10Mbit/sec Ethernet. The parameters of the genetic algorithm described in this experiment are shown in <Table 1>.

<Table 1> Parameters

	Size and rates
Population size	Variable size
The number of generation	Variable size (50-100)
Crossover rate	0.6
Mutation rate	0.1
Inversion rate	0.1

TSP instances are taken from the TSPLIB[11]. In the experiment, dynamic migration window size starts with 1, after the first migration window size increase 1. So, window size become 2, continue until window size is equal to 20% of population. The second method, whenever migration occurs, the dynamic migration window size varies at random from 1 to θ. The θ value is generated at random within 20% of the population size. And migration occurs per 30, 20 or 10 generations in order to reduce communication time rather than each generation. [Fig. 2] shows the results using variable migration rates.



[Fig. 2] The results using variable migration rates

6. Conclusion and Future works

TSP is a well-known combinatorial optimization problem and belongs to the class of NP-complete problems, various techniques are required for finding optimum or near optimum solution to the TSP.

This paper designs a distributed genetic algorithm in order to reduce the execution time and obtain more near optimal using multi-slave model for the TSP. Distributed genetic algorithms with multiple populations are difficult to configure because they are controlled by many parameters that affect their efficiency and accuracy. Among other things, one must decide the number and the size of the populations, the rate of migration, and the destination of the migrants. In this paper, I develop random dynamic migration rate that controls the size and the frequency of migration. Also I propose new migration policy that selects the destination of the migrants among the slaves.

In the future work, I will find more optimal dynamic migration rate, (, and more efficient migration policy to solve the TSP. Also, I will study the effect of migration on preventing premature convergence and will design more efficient distributed genetic algorithm for NP-complete problems.

※ References

[1] Erick Cantu´-Paz, David E. Goldberg, A Su mmary of Research on Parallel Genetic Alg orithms, *IlliGAL Report No. 95007*, 1995.

[2] A.Colorni, M. Dorigo, F. Maffioli, V. Mani ezzo, G. Righini, M. Trubian, Heuristics fro m Nature for Hard Combinatorial Optimizati on Problems, *Int. Trans. in Operational Res earch*, 3, 1, 1-21.

[3] C. Cotta, J.F. Aldana, A.J. Nebro, J.M. Tro ya, Hybridizing Genetic Algorithms with Br anch and Bound Techniques for the Resoluti on of the TSP, Springer-Verlag, *Artificial N eural Nets and Genetic Algorithms*, 1997, p p.277-279.

[4] http://www.cs.tcd.ie/research_groups/aig/iag/ar eab.html, "Distributed Problem Solving in M ulti-Agent Systems.

[5] David E. Goldberg, Genetic Algorithms: in Search and Optimization, *Addison-Wesley*, 1 998, pp.1-125.

[6] Terry Jones, Stephanie Forrest, Genetic Algo rithms and Heuristic Search, *International J oint Conference on Artificial Intelligence*, Ja nuary 1995.

[7] J. Kim and Y. Hong, "A Distributed Hybrid Algorithm for the Traveling Salesman Probl em", *Journal of KISS, Vol.25 No. 2*, 1998, pp. 136-144.

[8] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, *Springer-v erlag*, 1995, pp. 209-237.

[9] Gerhard Reinelt, The Traveling Salesman C omputational Solutions for TSP Applications, *Springer-Verlag*, 1994.

[10] T. C. Sapountzis, The Traveling Salesman P roblem, *IEEE Computing Futures*, Spring 19 91, pp. 60-64.

[11] TSPLIB, http://www.iwr.uni-heidelberg.de/iwr/ comopt/soft/TSPLIB95/ATSP.html.

[12] Erick Cantu´-Paz, "Designing Efficient Master-Slave Parallel Genetic Algorithms, *IlliGAL Technical Report No. 97004*, May 1997.

[13] Erick Cantu´-Paz, "A Survey of Parallel Genetic Algorithm", *IlliGAL Technical Report 97003*, May 1997

[14] Marco Tomassini, "Parallel and Distributed Evolutionary Algorithms : A Review", *http://www-iis.unil.ch*

김 정 숙

1993년 2월 동국대학교 컴퓨터공학과 졸업(공학사)
1995년 2월 동국대학교 대학원 컴퓨터공학과 졸업 (공학석사)
1999년 8월 동국대학교 대학원 컴퓨터공학과 졸업 (공학박사)
2000년 3월 ~ 현재 김포대학 컴퓨터계열 소프트웨어 개발 전공 전임강사