ISSN No: 2348-4845

Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm in Multicore Processors

Vandana Mahalakshmi, Assistant professor, AITAM, Tekkali,AP, vandanamahalakshmi@gmail.com

The traveling sales man problem is one of the mostly used real time problem. Many applications involves robot arm movement to fix the nuts in a large production environment, postal van services, shortest path finding technique in a communication environment can get more beneficial by implementing the TSP – PSO in multithreaded environment. It is one of the combinatorial problem that comes under NP-Hard problems. NP-Hard problems are the problems which can be solved by a nondeterministic turing machine in polynomial time, that is, NP-Problems consumes non-polynomial amount of time. Particle Swarm Algorithm is one type of algorithm used to solve traveling salesperson problem which is some what better than genetic algorithms in certain parameters. If the PSO algorithm is applied in multicore processors with multi threading approach the overall performance of the system can be improved.

Keywords: particle swarm optimization, travelling salesman, performance, multithread.

Introduction

The traveling sales man problem is one of the mostly used real time problem. Many applications involves robot arm movement to fix the nuts in a large production environment, postal van services, shortest path finding technique in a communication environment can get more beneficial with this TSP – PSO in multithreaded environment.[15] TSP is one of the challenging problem in operational-research. The description of TSP involves a tour of sales person who covers all cities only once of each with a shortest distance travelled and finally reached to the initial city. Here the cities are represented by a vertices of graph and the distance between any city is represented by the edge cost. The solution is a shortest path existed[12].

This paper introduce the PSO algorithm for TSP on multi core processors, we use the proposed algorithm for solving the combinatorial problem: TSP, the new algorithm shows great efficiency in solving TSP. the new algorithm is efficient than the genetic algorithm in comparing with the TSP[1]

Many real world optimization problems are time-dependent and some of them can be modeled by the dynamic TSPs (DTSPs). A DTSP is harder than a general TSP, which is a NP-hard problem, because the city number and the cost matrix of a DTSP are time varying. Although DTSP is a very common and important model in real world systems, few literatures have discussed this related issues. There are many open questions about DTSP urgently needed to be answered. We first give a mathematical model and the optimization objective for DTSP[4]. Then we discuss why evolutionary algorithms (EAs) are effective for solving DTSPs and give some key points for designing efficient DTSP EAs. By defining three dynamic operators, we proposed an evolutionary algorithm for DTSPs[5].

The main benefits that arise from multithreading are:

- improved application responsiveness and better program structure - any program in which many activities do not depend upon each other can be redesigned so that each activity is executed as a thread[10],
- efficient use of multiple processors - numerical algorithms and applications with a high degree of parallelism, such as matrix multiplication, can run

ISSN No: 2348-4845

much faster when implemented with threads on a multiprocessor[14]

- use fewer system resources - the cost of creating and maintaining threads is much smaller than the cost for processes, both in system resources and time[13].

A major source of speedup is the parallelizing of operations. Parallel operations can be either within-processor, such as with pipelining or having several ALUs within a processor, or between processosr, in which many processors work on diffrent parts of a problem in parallel. Our focus here is on between-processor operations [5].

Comparison of SIMD and MIMD

In contrast to MIMD systems, processors in SIMD|Single Instruction, Multiple Data|systems

execute in lockstep. At any given time, all processors are executing the same machine instruction on diffrent data[12].

Some famous SIMD systems in computer history include the ILLIAC and Thinking Machines

Corporation's CM-1 and CM-2. Also, DSP (\digital signal processing") chips tend to have an

SIMD architecture.

But today the most prominent example of SIMD is that of GPUs|graphics processing units. In

addition to powering your PC's video cards, GPUs can now be used for general-purpose computation[17].

The architecture is fundamentally shared-memory, but the individual processors do execute

in lockstep, SIMD-fashion.

Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) algorithm was an intelligent technology first presented in 1995 by Eberhart and Kennedy, and it was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities [5]. If we compare PSO with Genetic Algorithms (GAs), we may find that they are all maneuvered on the basis of population operated. But PSO doesn't rely on genetic operators like selection operators, crossover operators and mutation operators to operate individual, it optimizes the population through information exchange among individuals[15]. PSO achieves its optimum solution by starting from a group of random solution and then searching repeatedly.

Fitness function is the only standard of judging whether an individual is "good" or not. We take the reciprocal of the length of each path as the fitness function. Length the shorter, fitness values the better. The fitness function is defined as following formula:

f(Si)=d(Cn(1),Cn(2)) + d(Cn(2) +Cn(3))+ …….d(Cn(n-1)+d(Cn(n))+d(Cn(n),Cn(1))

This paper introduce the PSO algorithm for TSP on multi core processors, we use the proposed algorithm for solving the combinatorial problem: TSP, the new algorithm shows great efficiency in solving TSP. the new algorithm is efficient than the genetic algorithm in comparing with the TSP.

Matrix multiplication as an example for multithreaded environment

Matrix multiplication is a good example of linear multiplication where two matrices A of size MXN and B is of size NXP are multiplied and a result of C matrix of size MX P is generated.[2] As it needs the time complexity of O(n3) which is very large than compared to any other parallel methods . if the multithreading is applied to this method then performance will be increases largely as certain experiment results shows this case[20]

Each thread executes the same code which is outlined below:

```
procedure thread;
lock;
while lastrow < M do
my_row := lastrow;
lastrow++;
unlock;
for i:=1 to P do
C(my_row,i) := 0;
for j:=1 to N do
C(my_row,i) := C(my_row,i)+A(my_row,j) . B(j,i);
lock;
```

unlock;

In the same manner TSP-PSO algorithm can be implemented in multi core environment on a multithreaded base to improve the performance at certain level. If the threads are implemented on a low communication environment then results many not be good, where as if the threads are implemented in amore communication environment the overall performance will be improved for larger values of vertices(cities).[7]

wide application in real practice such as Path Problem, Routing Problem and Distribution Problem, it has attracted researchers of various domains to work for its better solutions. Those

traditional algorithms such as Cupidity Algorithm, Dynamic Programming Algorithm, are all facing the same obstacle, which is when the problem scale of lrger values of N.[9]

Conclusion

This paper introduces the TSP-PSO in multithreaded environment, under certain conditions the multithreading can improve the performance of given algorithms which are running on multiprocessor system. If threads run independently or with very low communication, speedup is only limited by the number of processors. If threads run on a high communication environment speed will be improved and overall system performance can be improved.

References

1 Xuesong Yan1, Can Zhang1 , Wenjing Luo1, Wei Li1, Wei Chen1".Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm" , IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012

2.Leonardo Jelenković, Goran Omrčen-Čeko, " experiments with multithreading in parallel computing"

3. Davis GPU, Multicore, Clusters and More, "Programming on Parallel Machines Norm" Matlo_ University of California,

4.Aimin Zhou, Lishan Kang and Zhenyu Yan, "Solving Dynamic TSP with Evolutionary Approach in Real Time", In Congress on Evolutionary Computation(CEC'03), 2003, pp.951-957.

5.J. Kennedy and R. C.Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks, 1995, pp.1942-1948.

6. SunSoft, (1994), Solaris 2.4: Multithreaded Programming Guide, Sun Microsystems, Mountain View,California.

7. Sohn, A. (1995), .Parallel N-ary Speculative Computation of Simulated Annealing., IEEE Transactions on Parallel and Distributed Systems, vol. 6, pp. 997-1005.

8. Zhangcan Huang, Xiaolin Hu and Siduo Chen, "Dynamic Traveling Salesman Problem based on Evolutionary Computation", In Congress on Evolution

9. Pang, W.; Wang, K.; Zhou, C. & Dong, L, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem", In: Proceedings of the Fourth International Conference on Computer and Information Technology, Wuhan, China, September 2004, pp.796-800.

10. Machado, T.R. & Lopes, H.S, "A hybrid particle swarm optimization model for the traveling salesman problem", In: Natural Computing Algorithms, Ribeiro, H.; Albrecht, R.F. & Dobnikar, A. (Eds.), 2005, pp.255-258.

11. Goldbarg, E.F.G; Souza, G.R. & Goldbarg, M.C, "Particle swarm for the traveling salesman problem", Proceedings of the EvoCOP 2006, Gottlieb, J. & Raidl, G.R. (Ed.), Lecture Notes in Computer Science, Vol. 3906, 2006, pp.99-110.

12. Parsopoulos, K.E., Vrahatis, M.N., "Particle swarm optimization method for constrained optimization problems",In: Kvasnicˇ ka, V. et al. (Eds), Proceedings of the second Euro-International Symposium on Computational Intelligence, Kosˇ ice, Slovakia, 2002, pp.214–220.

13. Fang, L.; Chen, P. & Liu, S, "Particle swarm optimization with simulated annealing for TSP", Proceedings of the 6th WSEAS International Conference on Artificial Intelligence,

Knowledge Engineering and Data Bases, Long, C. A.; Mladenov, V. M. & Bojkovic Z. (Eds.), 2007, pp.206-210.