ABSTRAK

Metode *hybrid Ant Colony Optimization* (ACO) dan *Genetic Algorithm* (GA) mempunyai performa yang cukup meyakinkan untuk menyelesaikan masalah strategi *caching* pada mekanisme *cached data offloading* dengan model *Knapsack Problem* 0/1 (KP01). *Data offloading* adalah mekanisme bongkar muat *cached data* ke dalam *cache memory*. Model masalah KP01 mempunyai tujuan memasukkan sebanyak mungkin *cached data* ke dalam *cache memory* sehingga memperoleh profit paling optimal. Profit KP01 diformulasikan menggunakan fungsi objektif (Fx) yang dicari nilai paling minimumnya. Metode *cyclic hybrid* ACO-GA dengan mekanisme pemilihan solusi berbasis *roulette wheel selection* (RWS) pernah digunakan untuk menyelesaikan optimasi *cached data offloading*. Namun metode ini tidak mampu menemukan solusi optimum global. Kemudian siklus *cylic* ACO-GA dengan urutan ACO→GA→ACO juga menyebabkan konsumsi waktu algoritme hibrida ini semakin besar. Masalah lain yang belum diselesaikan adalah munculnya *polusi cache* yang disebabkan karena tidak adanya pengelolaan terhadap *access recency*. *Access recency* didefinisikan sebagai akses suatu *cached data* yang baru saja dilayani oleh *cache memory*. Polusi *cache* yang tidak ditangani dengan baik dapat menurunkan kinerja *hit ratio* (HR). Penelitian ini mengusulkan metode *nested*-RWS (nRWS) yang mampu menyeimbangkan korelasi sifat stokastik pada *random uniform* [0-1]. Penelitian ini juga mengubah siklus *cyclic* ACO-GA menjadi *non-cyclic* ACO-GA yang dikemas dalam *framework* GENACO untuk efisiensi waktu. Kemudian masalah polusi *cache* diselesaikan dengan pengelolaan *access recency* menggunakan *framework* GENACO dan algoritme *Least Recently Used* (LRU), yang kemudian diberi nama LRU-GENACO. Usulan metode dalam penelitian ini telah disimulasikan dan diuji menggunakan (i) *private dataset* KP01 serta (ii) *public dataset* dari IRcache. Simulasi dilakukan dengan mengukur nilai fungsi objektif (Fx), HR dan *Latency Saving Ratio* (LSR). Kemudian dilakukan proses *benchmarking* terhadap enam algoritme *caching* lainnya. Hasil simulasi menunjukkan bahwa metode nRWS mampu memperoleh Fx=0,4350 sementara nilai Fx penelitian sebelumnya hanya 0,4374. Metode nRWS berhasil menyeimbangkan korelasi antara sifat stokastik *random uniform* [0-1] dengan probabilitas kumulatifnya, sehingga dapat menemukan solusi optimum global dengan nilai Fx yang lebih kecil. Perbaikan siklus *cyclic* menjadi *non-cyclic*, berhasil mengurangi konsumsi waktu hingga empat kali lebih cepat. Selanjutnya, usulan metode LRU-GENACO, berhasil meningkatan rata-rata nilai HR dari algoritme LRU, LFU, LFUDA, SIZE, GDS, dan GDSF masing-masing sebesar 1,004%, 8,56%, 3,11%, 9,52%, 5,8%, dan 2,63%. Hasil tersebut diperoleh dengan melakukan simulasi menggunakan *dataset* IRcache. Pola akses data berdistribusi normal pada *dataset* IRcache lebih menguntungkan bagi LRU-GENACO untuk meningkatkan kinerja HR. Kinerja HR dipengaruhi oleh *cache size*, *unique request*, dan *cacheable request*. Semakin besar *cache size* semakin besar kinerja HR. Semakin kecil *unique* dan *cacheable request* justru membuat kinerja HR semakin besar.

Kata kunci: *roulette wheel selection*, *cached data offloading*, *knapsack problem 0/1*, *ant colony optimization*, *genetic algorithm*, *least recently used*

ABSTRACT

The Ant Colony Optimization (ACO) and Genetic Algorithm (GA) hybrid method have a convincing enough performance to solve the caching strategy problem in the cached data offloading mechanism with the Knapsack Problem 0/1 model (KP01). Data offloading is a mechanism for loading and unloading cached data into the cache memory. The KP01 problem model aims to enter as much cached data as possible into the cache memory to obtain the most optimal profit. KP01 profit is formulated using the objective function (Fx) to find the minimum value. The cyclic hybrid ACO-GA method with a roulette wheel selection (RWS) based solution mechanism has been used to solve cached data offloading optimization. However, this method is not able to find the optimal global solution. Therefore, the ACO-GA cyclic cycle with the order ACO→GA→ACO causes the time consumption to increase. Another problem that needs to be resolved is the emergence of cache pollution caused by the lack of management of access recency. Access recency is access to cached data that has just been served by cache memory. Cache pollution that is not handled correctly can reduce the performance-hit ratio (HR). This study proposed a nested-RWS (nRWS) method that can balance the correlation of stochastic properties at random uniform [0-1]. This research also changed the cyclic ACO-GA cycle to non-cyclic ACO-GA, packaged in the GENACO framework for time efficiency. Then the cache pollution problem was solved by managing access recency using the GENACO framework and the Least Recently Used (LRU) algorithm, later named LRU-GENACO. The proposed method in this study has been simulated and tested using (i) private dataset KP01 and (ii) public dataset from IRcache. The simulation measures the value of the objective function (Fx), HR, and Latency Saving Ratio (LSR). Subsequently, the benchmarking process is carried out against six other caching algorithms. The simulation results show that the proposed nRWS method can obtain Fx=0.4350, while the previous research was only 0.4374. The proposed nRWS method successfully balances the correlation between the random uniform stochastic properties [0-1] with its cumulative probability to find a global optimum solution with a smaller Fx value. Subsequently, improving cyclic cycles to non-cyclic reduced time consumption up to 4 times faster. Furthermore, the proposed LRU-GENACO method increased the average HR value of the LRU, LFU, LFUDA, SIZE, GDS, and GDSF algorithms by 1.004%, 8.56%, 3.11%, 9.52%, 5.8%, and 2.63%, respectively. These results were obtained by performing a simulation using the IRcache dataset. The normally distributed data access pattern on the IRcache dataset is more beneficial for LRU-GENACO to improve HR performance. HR performance is affected by cache size, unique requests, and cacheable requests. The larger the cache size, the greater the HR performance. The smaller the unique and cacheable requests, the more excellent HR performance will be.

Keywords: roulette wheel selection, cached data offloading, knapsack problem 0/1, ant colony optimization, genetic algorithm, least recently used