Mining of High Utility Item sets using Genetic Algorithm

Vamsinath Javangula [a], Suvarna Vani Koneru [b] and Haritha Dasari [c]

[a] Research Scholar, Department of CSE, JNTU Kakinada, Andhra Pradesh, India
[b] Professor, Department of CSE, V. R Siddhartha Engineering College, Kanuru, Andhra Pradesh, India
[c] Professor , Department of CSE, University College of Engineering College, JNTU Kakinada, Andhra Pradesh, India

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: A high utility item set mining system using genetic algorithm has been proposed in this article. The proposed system mines high utility item sets by parsing the database only once. The high utility item set mining problem has been reformulated as a constrained optimization problem. Genetic algorithm has been used to mine the high utility item sets. The entities such as gene, chromosome, population and fitness function, required to apply genetic algorithm for mining high utility item sets has been defined. In the proposed system, due to the requirements imposed by the high utility item set mining problem, a non binary representation of the genetic algorithm has been proposed. The operators of genetic algorithm such as cross over and selection operation have been customized to efficiently operate in the mining problem. To reduce the explosion of candidate generation, an upper bound based on the sum of the item set utility and remaining utility is used.
The item set utility is compared with the threshold to select the high utility item sets. The remaining utility is used to select prospective candidates for superset generation. The algorithm starts with an initial population comprising of single item set. The fitness function computes the sum of the item set utility and the remaining utility and selects the prospective item sets for breeding. The children are generated by merging two prospective item sets. The selection and breeding process are repeated until there are no more chromosomes in the population for further breeding. The proposed system was able to identify high utility item sets in a single scan of the database. The genetic algorithm was able to converge from an initial population of item sets to high utility item sets. Extensive testing of the proposed system showed the optimal potential of the system verified to other similar state of the art system.

Keywords: High Utility Item Set Mining, Transaction Utility Mining, Item Set Mining, Utility Mining, Mining Weighted Frequent Patterns, Genetic Algorithm, Gene, Population, Chromosome, Fitness Function, Breeding Process

1. Introduction

High utility item set (HUI) mining involves identifying item sets in a transaction that yield high utility. It is a variation of the widely researched frequent item set mining problem. In case of frequent item set mining just the frequency of the item set in the transactions is taken as a parameter for indicating the importance of the item set. But in HUI mining, every item is related with some kind of utility measure which is then used for indicating the importance of the items.

In most cases the utility measure comprises of two components – external and internal. The external utility measure for an item is usually set and independent of the transactions involving the item. The internal utility measure of an item on the other hand is dependent on the transaction in which it is involved and hence may vary from transaction to transaction. The formal definitions of concepts involved in HUI mining is given in [Yao, et.al, 2004]. In this paper a simplified model (Table 1) of super market transaction database (TD) is used as a running example for illustrating the mining process.

Table 1: Super Market Transaction Database

Bill No.	**Quantity purchased**							**Bill Value**
	Item1 (1)	**Item2 (3)**	**Item3 (5)**	**Item4 (2)**	**Item5 (2)**	**Item6 (1)**	**Item7 (1)**	
B1	1	-	1	-	1	-	-	8
B2	6	2	2	-	-	5	-	27
B3	1	1	1	2	6	-	5	30
B4	3	1	-	4	3	-	-	20
B5	2	1	-	2	-	2	-	11

In Table 1, the cost per item is given within parenthesis below the item label i.e. the cost per item of item 3 is 5. Each row in the transaction contains the quantity of items purchased and the total value of the transaction in the "Bill Value" column. The transactions are uniquely identified by the "Bill No.". For example, the value of bill "B3" is 30. In the terminology of HUI mining, the cost per item is the external utility of an item and the quantity of item purchased against each bill is the internal utility.

The utility of each item in a transaction is the multiplication of the quantity of item involved and cost per item. For example,

Utility of item4 in transaction "B4" = 4 x 2 = 8.

The utility of an item set in a transaction is obtained by summing up the utilities of the item set in the transaction. For example,

Utility of {item3, item4} in "B3" = 5 + 4 = 9.

The utility of a transaction is computed by adding up the utilities of all the items involved in the transaction. For example,

Utility of "B5" = 2 + 3 + 4 + 2 = 11.

The utility of each item in TD is the sum of the utilities of the item in each transaction in TD in which it is present. For example,

Utility of item6 in TD = utility of item6 in B2 + utility of item6 in B5 = 5 + 2 =7.

The utility of an item set in TD is the sum of the utilities of the item set in each transaction in TD in which it is present. For example,

Utility of {item2, item6} in TD = utility of {item2, item6} in B2 + utility of {item2, item6} in B5

= 11 + 5 = 16.

The utility table corresponding to the TD in Table 1 computed as above is presented in Table 2. The transaction utility feature of the table indicates the overall utility of the transaction. The item utility row in the table shows the utility of every item in the transaction database.

Table 2: Utility Table for TD

Bill No.	**Item1**	**Item2**	**Item3**	**Item4**	**Item5**	**Item6**	**Item7**	**Transaction Utility**
B1	1	-	5	-	2	-	-	8
B2	6	6	10	-	-	5	-	27
B3	1	3	5	4	12	-	5	30
B4	3	3	-	8	6	-	-	20
B5	2	3	-	4	-	2	-	11
Item Utility	13	15	20	16	20	7	5	

In this article, HUI mining using genetic algorithm (GAHUIM) has been proposed. The sections following the introduction are – Section 2 Literature Survey, Section 3 GAHUIM the proposed system, Section 4 Experimentation and results and Section 5 Conclusion and Future Scope.

2. BACKGROUND

In this section a brief of the various developments and modern systems in the research area of high utility item set mining [Yao, et.al, 2004]. The earlier researches led to development of systems that followed approaches similar to Apriori algorithm [Agrawal and Srikant, 1994] used for frequent item set mining. These systems [Barber and Hamilton, 2000, Li, et.al, 2005a] systematically start with scanning the database and pick 1-itemsets based on the transaction weighted utility [Liu, et.al, 2005] of the item sets.. Then, the database scanned a second time to estimate the actual utility of these item sets and prune item sets not satisfying the minimum utility threshold.

The above process is repeated until all higher order item sets have been identified. These approaches employed two database scan in each iteration to calculate the actual utility of the item sets. This was improved in [Li, et.al, 2005b] where the real utility of the item sets were computed in a single scan of the database itself. This was followed by other approaches coming up with novel data structure and pruning strategies to build more efficient systems. In [Dam, et.al, 2019] a closed HUI mining system has been projected which uses new pruning methods such as chain estimated utility co-occurrence pruning, lower branch pruning, and pruning by coverage for better results.

A novel utility measure and a new data structure – pset and a new algorithm has been developed in [Nguyen, et.al, 2019] for mining HUIs in databases with item sets showing dynamic profits. In [Uday Kiran, et.al, 2019], HUIs which are occurring more frequent are mined, using a cutoff utility, threshold support, and suffix utility measures. [Duong, et.al, 2018] has proposed certain modifications to utility list data structure to facilitate better memory management and join operations. Later, to get better the efficiency of the mining systems FP-Growth [Han, et.al, 2004] based methods were employed.

These approaches [Ahmed, et.al, 2009, Tseng, et.al, 2010, Tseng, et.al, 2013] utilized a prefix tree structure to hold pattern and utility information from the database. The prefix tree is then traversed recursively to generate the HUIs. To further speed up the mining process, some approaches [Lan, et.al, 2014, Qu, et.al, 2019, Yun, et.al, 2014] utilized customized novel data structures such as maximum item quantity tree, indices, utility lists etc. Other approaches experiment with algorithms simulating natural optimization processes.

In [Kannimuthu and Premalatha, 2014] an approach for mining HUIs with negative utility values and no user specified threshold using genetic algorithm has been proposed. [Arun Kumar, et.al, 2018] have proposed a mining strategy using ant colony algorithm. Irrespective of all these developments, it was still found that there was scope for further speeding up the operation by consuming lesser memory space.

3. GAHUIM

Genetic algorithm (GA) is an optimization algorithm mimicking natural evolution [Eiben, et.al, 2003]. In general GA performs a search over a space of solutions and picks the optimal solutions. Figure 1 shows the basic processes and flow of GA.

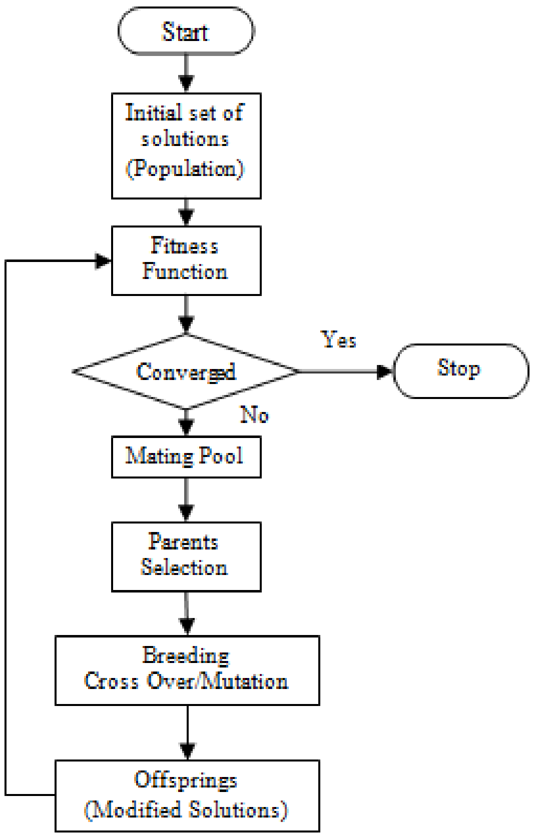

Start
Initial set of solutions (Population)
Fitness Function
Converged
Yes
Stop
No
Mating Pool
Parents Selection
Breeding Cross Over/Mutation
Offsprings (Modified Solutions)

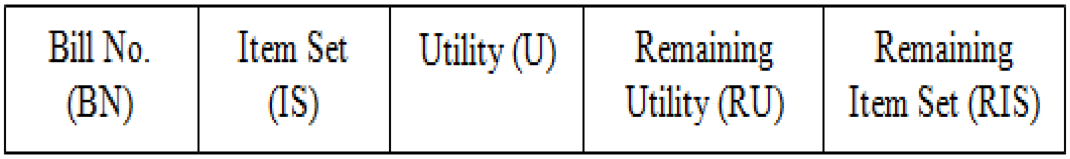
Figure 1. GA processes and flow

The fitness function is decided upon the requirements of the problem for which the solution is sought for. It tries to pick good solutions from the initial population and subjects it to modification by applying cross over and/or mutation. The new solutions (offsprings) thus obtained is added to the population. Thus, the solutions from the initial population are gradually refined to move it towards the optimal solution.

The process comes to an end if the population converged to a population representing optimal solution.In order to use GA, the HUI mining problem has to be modeled as an optimization problem. The HUI mining technique can be considered as a constrained optimization problem where maximization of utility with respect to the items involved in the transactions need to be done. The remodeling of HUI mining problem for GA involves the following entities:

3.1 Gene

A single dimensional array containing information about an item set in a transaction in TD. Figure 2 shows the gene structure. Due to the nature of the problem domain, a non binary form of representation is used for the model of GA.



Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)

Figure 2. Structure of a the Gene

In **Figure 2**, the components making up the gene are as follows:

BN – Bill No. of the transaction under consideration.

IS – The item set under process in BN.

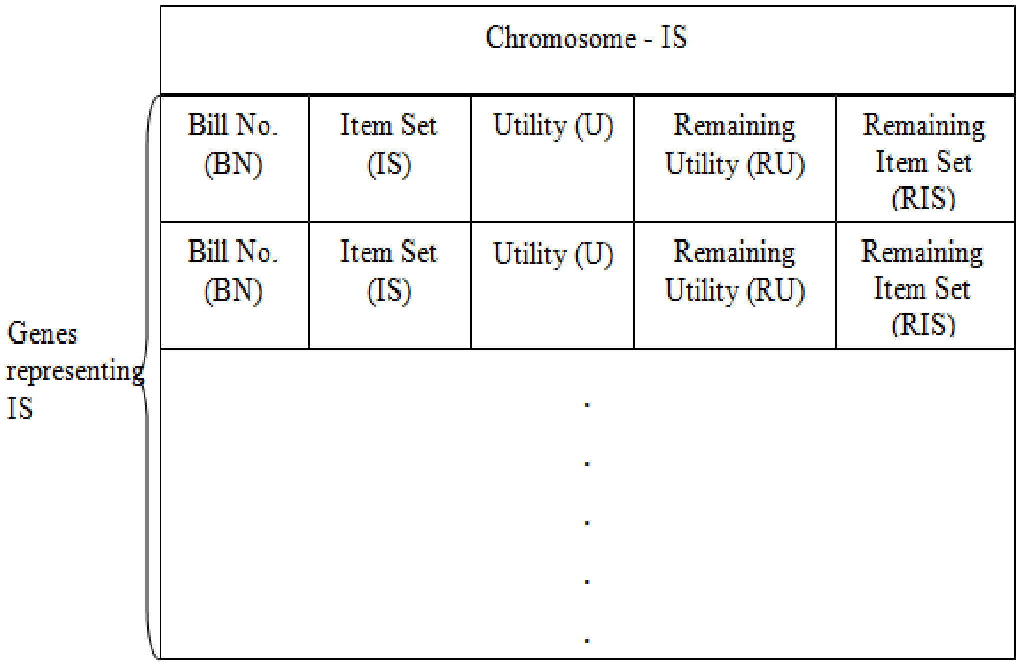
U – Utility of IS in BN.

RU – The utility sum of items following IS in BN.

RIS – The items following IS in BN.

3.2 Chromosome

A set of genes representing an item set in TD.



Chromosome - IS				
Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)
Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)
. . . . .				

Figure 3. shows the structure of a chromosome.

3.3 Population

A set of chromosomes. Each chromosome in the set represents an item set in TD.

3.4 Fitness Function

In the HUI mining problem, two functions are used for fitness selection – HUISelect and MatSelect.

- **HUISelect** – Copies chromosomes representing HUIs to output. The chromosomes whose utility value is upper than a user specified threshold are taken as HUIs. The utility of a chromosome is The genes utilities sum which making up the chromosome.
- **MatSelect** – Chooses chromosomes with higher probability of generating HUIs to mating pool. The mating pool comprises of two sections primary and secondary. All the chromosomes in the initial population are transferred to the secondary section. If the summation of remaining utility and utility of a chromosome is greater than the user defined threshold, then it is transferred to the primary section of the mating pool.

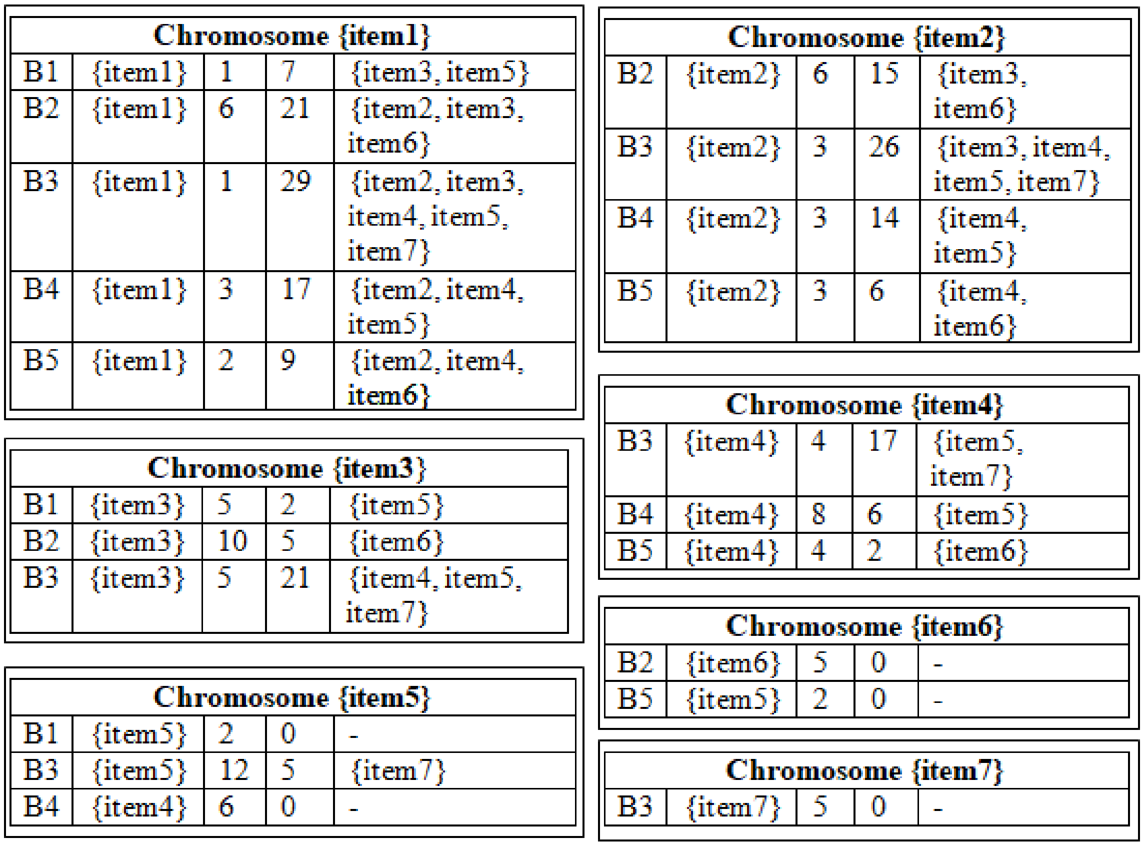
3.5 Breeding Process

Two chromosomes are merged to form an offspring. A chromosome from the primary section can only be merged with another chromosome in the secondary section whose item set is part of the RIS of the primary chromosome. The utility of the offspring is the summation of the utility of its parent and RU of the offspring is the lowest of the RU of its parent. The RIS of the gene of the offspring is the RIS of the matching gene (same BN) of the secondary chromosome. After generating all the offspring of the primary chromosome, it is discarded.

3.6 Convergence

Provides the termination condition. In the HUI mining problem, the process is iterated n times, where n is the no. of items in TD i.e. if no. of items in TD is 5 then the process is iterated 5 times.
The GA process starts with the formation of the initial population. The initial population is built by scanning the TD and forming 1-item set i.e. item set comprising of a single item. To facilitate the computational processes a lexicographic ordering is imposed on the items scanned from TD i.e. item1≺item2≺item3≺…≺itemn, where n is the no. of items in TD. For example, Figure 4 shows the initial population of TD.
The fitness of the initial population is then checked using the HUISelect and MatSelect functions. For example, let the user defined threshold UTIL be 20. The HUISelect function copies the high utility chromosomes from the initial population to HUI set. As it can be seen, only two chromosomes – {item3} and {item5} – have a utility of 20 and are copied to HUI set.

The MatSelect function then chooses the chromosomes for the primary section of the mating pool by comparing U+RU values of it with the UTIL.Since the U+RU values of chromosomes {item6} and {item7} are less than UTIL they are discarded and the others are moved to the primary section. All the chromosomes in the initial population are moved to the secondary section of the mating pool. This is illustrated in the Figure 5.
In the breeding phase, the chromosomes from the primary section in the mating pool are picked up one by one and the offspring are generated. The mating pairs are picked from the secondary chromosome. The mating secondary chromo somes are identified by the items in the RIS of the primary chromosome.



Chromosome {item1}				
B1	{item1}	1	7	{item3, item5}
B2	{item1}	6	21	{item2, item3, item6}
B3	{item1}	1	29	{item2, item3, item4, item5, item7}
B4	{item1}	3	17	{item2, item4, item5}
B5	{item1}	2	9	{item2, item4, item6}

Chromosome {item2}				
B2	{item2}	6	15	{item3, item6}
B3	{item2}	3	26	{item3, item4, item5, item7}
B4	{item2}	3	14	{item4, item5}
B5	{item2}	3	6	{item4, item6}

Chromosome {item3}				
B1	{item3}	5	2	{item5}
B2	{item3}	10	5	{item6}
B3	{item3}	5	21	{item4, item5, item7}

Chromosome {item4}				
B3	{item4}	4	17	{item5, item7}
B4	{item4}	8	6	{item5}
B5	{item4}	4	2	{item6}

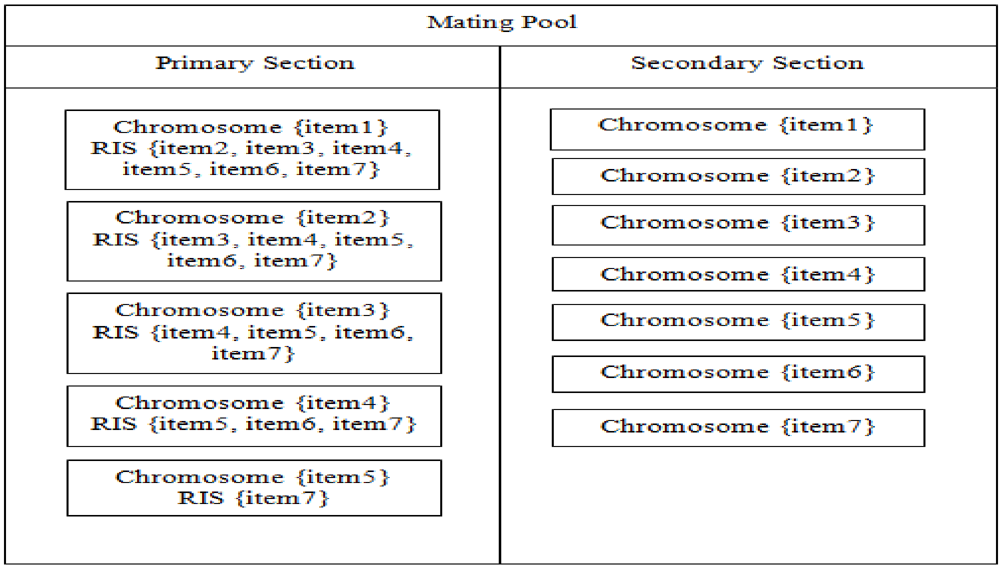
Chromosome {item5}				
B1	{item5}	2	0	-
B3	{item5}	12	5	{item7}
B4	{item4}	6	0	-

Chromosome {item6}				
B2	{item6}	5	0	-
B5	{item5}	2	0	-

Chromosome {item7}				
B3	{item7}	5	0	-

Figure 4. Initial Population

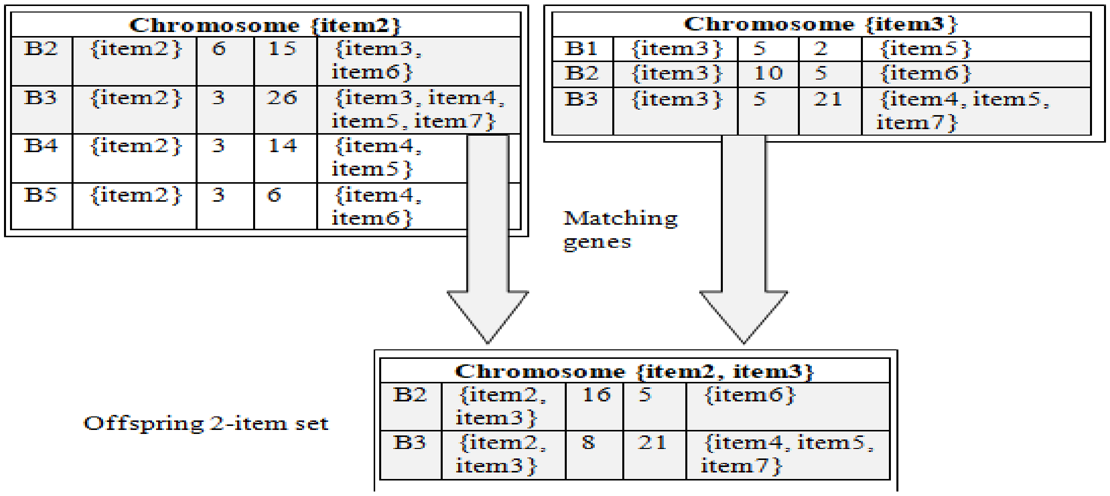
For example, as seen in Figure 5, the mating chromosomes of chromosome {item2} are chromosomes {item3, item4, item5, item6, item7}. Likewise, the mating chromosome of chromosome {item5} is chromosome {item7} as it is the only chromosome listed in the RIS of chromosome (item5}.



Mating Pool	
Primary Section	Secondary Section
Chromosome {item1} RIS {item2, item3, item4, item5, item6, item7}	Chromosome {item1}
Chromosome {item2} RIS {item3, item4, item5, item6, item7}	Chromosome {item2}
Chromosome {item3} RIS {item4, item5, item6, item7}	Chromosome {item3}
Chromosome {item4} RIS {item5, item6, item7}	Chromosome {item4}
Chromosome {item5} RIS {item7}	Chromosome {item5}
	Chromosome {item6}
	Chromosome {item7}

Figure 5. Mating Pool

The primary section of the mating pool varies with the generation of offspring where as the secondary section remains fixed. The mating and offspring generation process for chromosome {item2} and {item3} is represented in Figure 6.



Chromosome {item2}				
B2	{item2}	6	15	{item3, item6}
B3	{item2}	3	26	{item3, item4, item5, item7}
B4	{item2}	3	14	{item4, item5}
B5	{item2}	3	6	{item4, item6}

Chromosome {item3}				
B1	{item3}	5	2	{item5}
B2	{item3}	10	5	{item6}
B3	{item3}	5	21	{item4, item5, item7}

Chromosome {item2, item3}				
B2	{item2, item3}	16	5	{item6}
B3	{item2, item3}	8	21	{item4, item5, item7}

Figure 6. Offspring generation process

The above process is repeated 7 times as the number of items in TD is 7. After completion, the HUI set will contain the high utility item sets in TD.

4. EXPERIMENTATION RESULTS

The proposed system was experimented on a pc running on intel i3 processor with 3GB RAM. The experimentation was conducted using 6 real world bench mark datasets – Accident, Chess, Mushroom , Retail available in FIMI Repository (FIMI) [http://fimi.ua.ac.be/, 2012] ,Chain [Pisharath, et.al, 2012] and Foodmart [Qu, et.al, 2019], and. The datasets and their characteristics are shown in Table 3.

Table 3: Bench Mark Datasets

Dataset	#Transactions	#Items	Avg. Transaction Length
Accidents	340183	468	33.8
Chain	1112949	46086	7.3
Chess	3196	75	36
Foodmart	55624	1559	4.5
Mushroom	8124	120	23
Retail	88162	16470	10.3

Since, the system was capable of identifying all HUIs, the main concern was the time taken and the system's memory requirements. Thus, the experimentations were done to identify the systems speed of operation and memory requirement for each of the datasets for varying utility thresholds. Figures 7 – 12 highlight the performance of the proposed system vis-à-vis best of the HUIs [Ahmed, et.al, 2009, Qu, et.al,2019, Tseng, et.al, 2010, Tseng, et.al, 2013].

Figure 7a Accident Runtime Performance

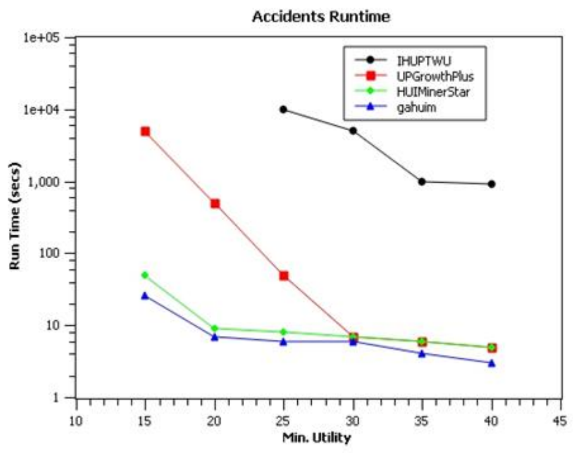


Figure 7b Accident Memory Usage

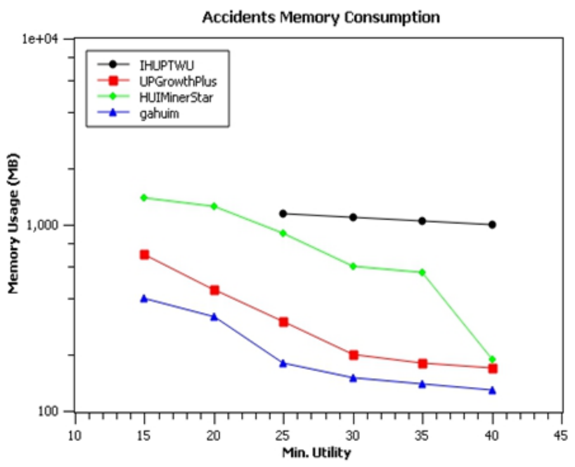


Figure 8a Chain Runtime Performance

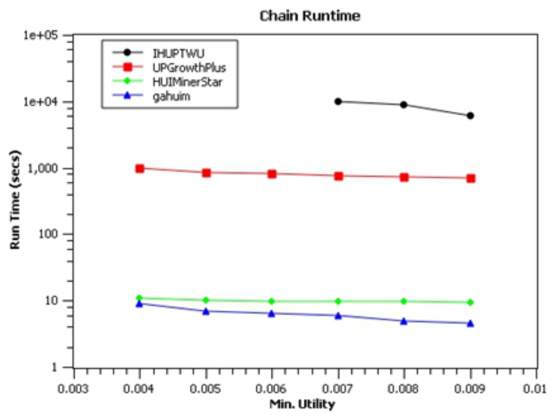


Figure 8b Chain Memory Usage

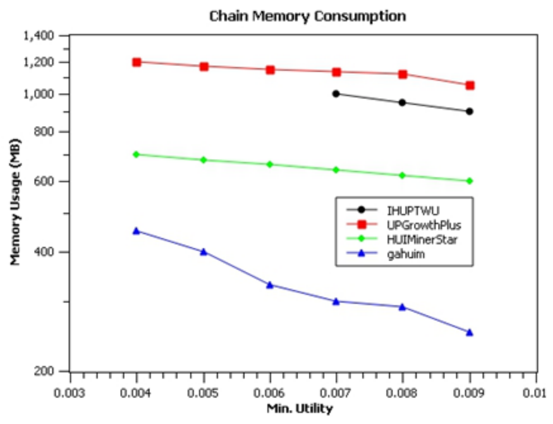


Figure 9a Chess Runtime Performance

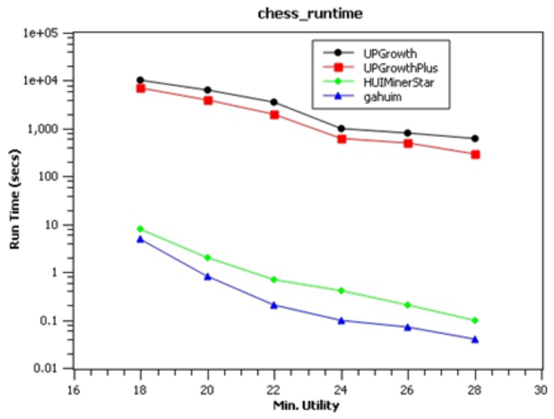


Figure 9b Chess Memory Usage

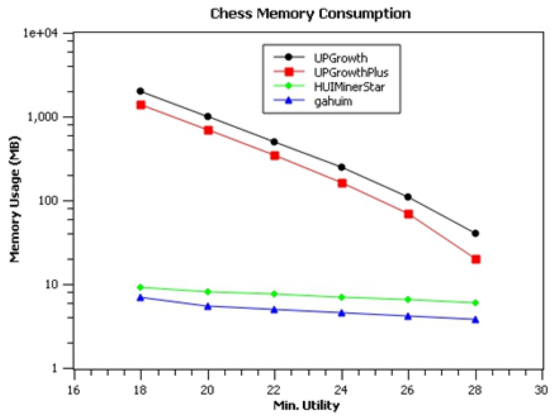


Figure10a Foodmart Runtime Performance

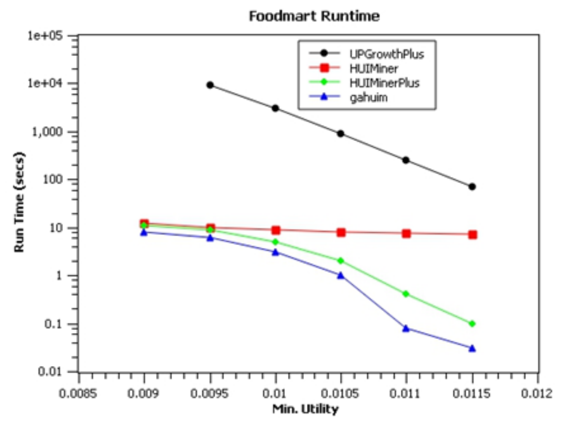


Figure 10b Foodmart Memory Usage

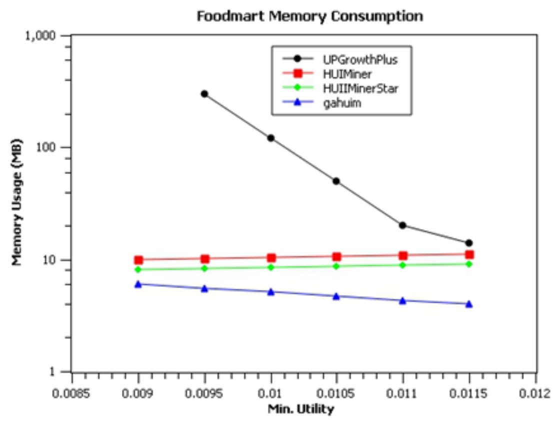


Figure 11a Mushroom Runtime Performance

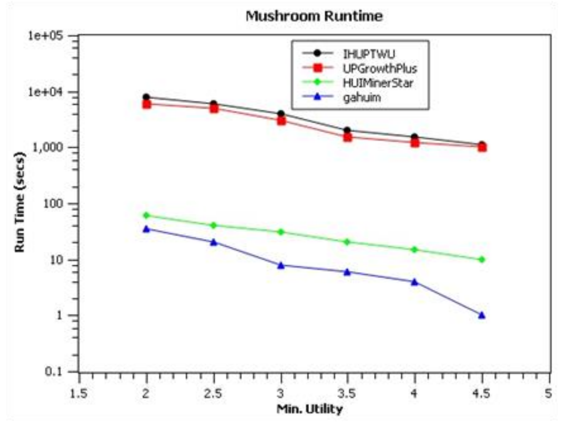


Figure 11b Mushroom Memory Usage

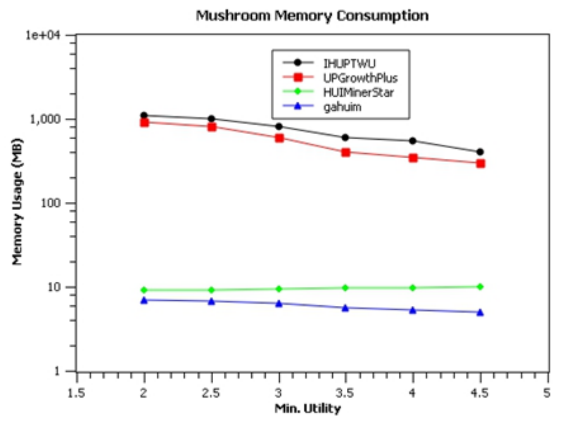


Figure 12a Retail Runtime Performance

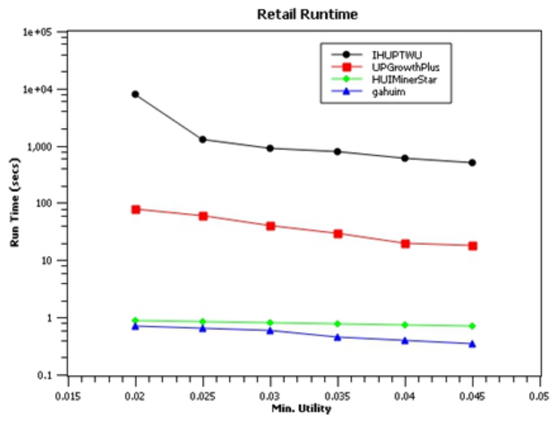
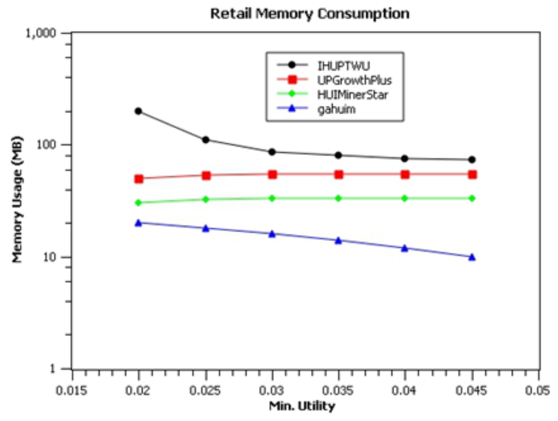


Figure 12b Retail Memory Usage



As can be seen, the proposed exhibited superior performance in terms of operational speed as well as memory requirements compared with other current HUI mining systems.

5. CONCLUSION AND FUTURE SCOPE

In this article, a HUI mining system using GA has been proposed. The HUI mining problem has been remodeled as a constrained optimization problem and GA has been customized to handle HUI mining problem. A non binary representation of GA has been used to mine the item sets. From an initial population of single item sets, prospective parents are identified based on the summation of the remaining utility and item set utility of the item set.
The children are reproduced using cross over function which combines two prospective parents to generate the new offspring. The selection and reproduction processes are iterated till no more parents are left for breeding. The system was tested with real bench mark datasets and was found to outperform the other current HUI mining systems. In future research, other representational forms such as binary and heuristics can be experimented with in the modeling of GA.

References

1. Agrawal, R., Srikant, R. (1994). *Fast algorithms for mining association rules in large databases. In: Proceedings of the International Conference on Very Large Data Bases, pp. 487–499.*
2. Ahmed, C. F., Tanbeer, S. K., Jeong, B. S., & Lee, Y. K. (2009). *Efficient tree structures for high utility pattern mining in incremental databases.IEEE Transactions on Knowledge and Data Engineering, 21(12), 1708-1721.*
3. Arunkumar, M.S., Suresh, P. & Gunavathi, C. (2018). *High Utility Infrequent Itemset Mining Using a Customized Ant Colony Algorithm. Int J Parallel Prog. doi: 10.1007/s10766-018-0621-7*
4. Aroulanandam, V.V., Latchoumi, T.P., Bhavya, B., Sultana, S.S. (2019). Object detection in convolution neural networks using iterative refinements. Revue d'Intelligence Artificielle, Vol. 33, No. 5, pp. 367-372. https://doi.org/10.18280/ria.330506
5. Balamurugan, K., Uthayakumar, M., Ramakrishna, M. and Pillai, U.T.S., 2020. Air jet Erosion studies on mg/SiC composite. Silicon, 12(2), pp.413-423.
6. Balamurugan, K., 2020. Compressive Property Examination on Poly Lactic Acid-Copper Composite Filament in Fused Deposition Model–A Green Manufacturing Process. Journal of Green Engineering, 10, pp.843-852.
7. Barber, B., & Hamilton, H. J. (2000). *Algorithms for Mining Share Frequent Itemsets Containing Infrequent Subsets. Principles of Data Mining and Knowledge Discovery Lecture Notes in Computer Science, 316–324. doi: 10.1007/3-540-45372-5_31*
8. Bhasha, A.C. and Balamurugan, K., 2020, July. Multi-objective optimization of high-speed end milling on Al6061/3% RHA/6% TiC reinforced hybrid composite using Taguchi coupled GRA. In 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE) (pp. 1-6). IEEE.
9. Dam, T., Li, K., Fournier-Viger, P. et al. (2019). *CLS-Miner: efficient and effective closed high-utility itemset mining. Front. Comput. Sci. 13, 357–381. Doi: 10.1007/s11704-016-6245-4*
10. Deepthi, T. and Balamurugan, K., 2019. Effect of Yttrium (20%) doping on mechanical properties of rare earth nano lanthanum phosphate (LaPO4) synthesized by aqueous sol-gel process. Ceramics International, 45(15), pp.18229-18235.
11. Duong, Q.-H., Fournier-Viger, P., Ramampiaro, H., Nørvåg, K., & Dam, T.-L. (2017). *Efficient high utility itemset mining using buffered utility-lists. Applied Intelligence, 48(7), 1859–1877. doi: 10.1007/s10489-017-1057-2*
12. Eiben, Agoston E., and James E. Smith. (2003). *Introduction to evolutionary computing. Heidelberg: Springer.*
13. *Frequent itemset mining dataset repository (2012). http://fimi.ua.ac.be/*
14. Garikipati P., Balamurugan K. (2021) Abrasive Water Jet Machining Studies on AlSi$_7$+63%SiC Hybrid Composite. In: Arockiarajan A., Duraiselvam M., Raju R. (eds) Advances in Industrial Automation and Smart Manufacturing. Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-15-4739-3_66
15. Gowthaman, S., Balamurugan, K., Kumar, P.M., Ali, S.A., Kumar, K.M. and Gopal, N.V.R., 2018. Electrical discharge machining studies on monel-super alloy. Procedia Manufacturing, 20, pp.386-391.

16. Han, J., Pei, J., Yin, Y., Mao, R. (2004). *Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min. Knowl. Discov. 8(1), 53–87.*
17. Kiran, R. U., Reddy, T. Y., Fournier-Viger, P., Toyoda, M., Reddy, P. K., & Kitsuregawa, M. (2019). *Efficiently Finding High Utility-Frequent Itemsets Using Cutoff and Suffix Utility. Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science, 191–203. doi: 10.1007/978-3-030-16145-3_15*
18. Kannimuthu, S., & Premalatha, K. (2014). *Discovery of High Utility Itemsets Using Genetic Algorithm with Ranked Mutation. Applied Artificial Intelligence, 28(4), 337–359. doi: 10.1080/08839514.2014.891839*
19. Lan, G. C., Hong, T. P., & Tseng, V. S. (2014). *An efficient projection-based indexing approach for mining high utility itemsets. Knowledge and information systems, 38(1): 85-107.*
20. Li, Y.-C., Yeh, J.-S., Chang, C.-C. (2005). *A fast algorithm for mining share-frequent itemsets. In: Proceedings Asia-Pacific Web Conference, pp. 417–428.*
21. Li, Y.-C., Yeh, J.-S., Chang, C.-C. (2005b). *Efficient algorithms for mining share-frequent itemsets. In: Proceedings of the 11th World Congress of International Fuzzy Systems Association, pp. 534–539.*
22. Liu, Y., Liao, W.-K., Choudhary, A. (2005). *A fast high utility itemsets mining algorithm. In: Proceedings of the Utility-Based Data Mining Workshop, pp. 90–99*
23. Nguyen, L. T., Nguyen, P., Nguyen, T. D., *Vo, B., Fournier-Viger, P., & Tseng, V. S. (2019). Mining high-utility itemsets in dynamic profit databases. Knowledge-Based Systems, 175, 130–144. doi: 10.1016/j.knosys.2019.03.022*
24. Pisharath, J., Liu, Y., et al. (2012). *NU-Minebench: a data mining benchmark suite*
25. Qu, J.-F., Liu, M., & Fournier-Viger, P. (2019). *Efficient Algorithms for High Utility Itemset Mining Without Candidate Generation. Studies in Big Data, 131–160. doi: 10.1007/978-3-030-04921-8_5*
26. Ranjeeth, S., Latchoumi, T.P., Sivaram, M., Jayanthiladevi, A. and Kumar, T.S., 2019, December. Predicting Student Performance with ANNQ3H: A Case Study in Secondary Education. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 603-607). IEEE.
27. Tseng, V. S., Shie, B. E., Wu, C. W., & Philip, S. Y. (2013). Efficient algorithms for mining high utility itemsets from transactional databases. IEEE transactions on knowledge and data engineering, 25(8), 1772-1786.
28. Tseng, V. S., Wu, C. W., Shie, B. E., & Yu, P. S. (2010). UP-Growth: an efficient algorithm for high utility itemset mining. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining ACM: 253-262.
29. Yao, H., Hamilton, H. J., & Butz, C. J. (2004). *A foundational approach to mining itemset utilities from databases." In Proceedings of the 2004 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics: 482-486.*
30. Yookesh, T.L., Boobalan, E.D. and Latchoumi, T.P., 2020, March. Variational Iteration Method to Deal with Time Delay Differential Equations under Uncertainty Conditions. In 2020 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 252-256). IEEE.
31. Yun, U., Ryang, H., & Ryu, K. H. (2014). *High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates. Expert Systems with Applications, 41(8), 3861-3878.*