INTERNATIONAL REVIEW OF ECONOMICS AND MANAGEMENT

ISSN: 2148-3493 IREM

DOI: 10.18825/iremjournal.272727 Volume 5, Number 1, 2017, 1-17.

A DECISION SUPPORT SYSTEM BASED ON GENETIC ALGORITHM FOR VARIABLE SIZED BIN PACKING PROBLEM WITH ITEM CONFLICTS

İnanç KABASAKAL[*]

Fatma Demircan KESKİN[†]

Abstract

Bin packing problem (BPP) is a combinatorial NP-hard problem that has variations including one, two and three dimensional packing, variable sized packing and packing with constraints. In the literature, exact and approximation algorithms have been mostly used to solve bin packing problems. Genetic Algorithms are meta-heuristic methods that have been applied to a vast majority of well-known optimization problems including the bin packing problems. In this paper, a variant of bin-packing problem for variable bins is addressed. The capacity constraints including volume and weight are given; moreover, to avoid item conflicts is defined as an additional constraint. A decision support model utilizing the genetic algorithm is introduced for this variant of the BPP. The performance of the model is tested with sample input, the results obtained are presented and discussed in the results section.

Keywords: Variable sized bin-packing problem, Genetic Algorithm, item conflicts, decision support systems

JEL Codes: C02, C6, M11

I. INTRODUCTION

Bin Packing Problem (BPP) is one of the classical combinatorial optimization problems and has received a great deal of attention in the literature due to its various application areas. In the classical BPP, a set of items must be assigned to a finite number of bins of fixed capacity in a way

[*] Ege University, Department of Business Administration, inanc.kabasakal@ege.edu.tr

[†] Ege University Department of Business Administration, fatma.demircan@ege.edu.tr

Date of submission: 05.12.2016

Date of acceptance: 20.06.2017

that minimizes the number of bins used. Besides classical one-dimensional BPP, there are many variations of BPP such as variable sized (Friesen & Langston, 1986; Kang & Park, 2003; Haouari & Serairi, 2009), two-dimensional (Li, Liu, Wu, & Xu, 2014), three dimensional (Martello, Pisinger, & Vigo, 2000), online (Kinnersley & Langston, 1988) and so on.

The problem addressed in this paper includes different categories of bins and each one contains limited and/or unlimited number of identical bins with the same capacity and cost. In addition, it contains different item categories; each involves finite number of identical and/or non-identical items. As we consider variable sized bin categories, our problem falls into the variable sized bin-packing problem (VSBPP) class with additional constraints that are considering volume capacities of bins in addition to weight capacities and conflicts between items.

VSBPP was first investigated by Friesen & Langston (1986). VSBPP is a generalization of the classical BPP and formally defined (Correia, Gouveia, & Saldanha-da-Gama, 2008) as follows: Given a set J of n items where each item j ($1 \leq j \leq n$) has a weight w_j and m different bin types. Each bin type k (k=1, …, m) includes an infinite number of identical bins, each having a capacity W_k and a fixed cost c_k. The objective is to minimize the total cost of bins used to pack all items.

Even though the classical BPP has been widely investigated in the literature, the VSBPP has received scant attention (Haouari & Serairi, 2011). Kang & Park (2003) addressed the VSBPP with two greedy algorithms they proposed to solve three particular cases of the problem. Correia et al. (2008) studied the VSBPP using a discretized model and obtained good linear programming bounds to solve instances including up to 1000 items. Bang-Jensen & Larsen (2012) investigated real life instances of the VSBPP with a construction and a local search heuristic with the aim of obtaining high quality solutions for large number of bin sizes and adapting fast to online changes in data. Maiza, Labed, & Radjef (2013) analyzed VSBPP with the aim of minimizing the total cost of bins by applying four heuristics they proposed. Dokeroglu & Cosar (2014) analyzed the one-dimensional BPP and applied bin-oriented heuristics in island parallel grouping genetic algorithms.

In the literature, there have also been studies addressing the BPP and VSBPP with item conflicts. Jansen (1999), Epstein & Levin (2008), Sadykov & Vanderbeck (2013) and Bodis (2015) have studied the BPP with conflicts. Epstein, Favrholdt, & Levin (2011) analyzed the online VSBPP with conflicts between items, which express that if there is a conflict on a pair of items, they cannot be assigned to the same bin and investigated online algorithms using mainly the asymptotic competitive ratio.

The BPP is an NP-hard problem (Korte & Vygen, 2006) and thus, the VSBPP is NP-hard as well. In the literature, exact and approximation algorithms are used to solve this problem. Monaci (2003) applied a branch-and-bound procedure to solve instances with 3 or 5 types of bins and up to 500 items. Alves and Valerio de Carvalho (2008) presented a branch-and-price-and-cut algorithm and used the instances of Belov & Scheithauer (2002) and Monaci (2003). Haouari & Serairi (2009) proposed a branch and bound algorithm to solve the test instances including three bin types, up to 500 items and randomly generated weights from discrete uniform distribution. Since the VSBPP is NP-hard these exact methods might require excessive computing time to solve large-scale instances. Many heuristic and meta-heuristic algorithms have been applied to the BPP and its extensions in the literature. In this paper, we proposed a genetic algorithm (GA) to solve the analyzed VSBPP.

GA is one of the most popular meta-heuristics that has been applied to a vast majority of well-known optimization problems including the BPP and its variants successfully. Falkenauer (1996) addressed the BPP by applying the Grouping GA of Falkenauer (1994). The Grouping GA is a group-oriented approach and it focuses on bins instead of items. Reeves (1996) used an approach hybridizing the GA with existing simple heuristics to solve the BPP. Iima & Yakawa (2003) proposed a new design of GA to solve one-dimensional BPP focusing on the item combinations in a bin. They designed a GA in a way that offspring inherit the combination of items. They solved 1210 benchmark instances to test the performance of their GA. According to their results, their GA performed better than Variable Neighborhood Search and BISON that is an approach combining the tabu search method and the branch-and-bound method in terms of accuracy of solutions. Mohamadi (2010) addressed the one dimensional BPP applying the GA with a new representation scheme. This scheme was constructed by combining the strong features of bin-based, object-based and group-based representation schemes. Quiroz-Castellanos, Cruz-Reyes, Torres-Jimenez, Gomez, Huacuja, & Alvim (2015) handled the one dimensional BPP with a method based on a grouping GA. Their method, which is referred to as the Grouping Genetic Algorithm with Controlled Gene Transmission, aimed the transmission of the best genes in the chromosomes with balancing the selective pressure and population diversity. Their approach improved the performance of grouping GA and was comparable to the best state of the art algorithms.

The paper is organized as follows. In Section 2, we present the formulation of the problem. Our solution algorithm is given in Section 3. Results of computational experiments are provided in Section 4. In Section 5, we discuss the results and draw conclusions.

II. FORMULATION OF THE PROBLEM

The problem analyzed can be defined as a VSBPP with conflicts including assignment of items to the variable sized bin categories considering weight and volume capacity of heterogenous bins while avoiding co-existence of conflicting item categories. The objective is to minimize the total cost of bins used to assign all items.

Indices, parameters, variables and formulation of the model are as follows:

Indices:

j, i= 1,2,…,n	Index of items
m, r =1,2,…,M	Index of item categories
k=1,2,…,K	Index of bin types

Parameters:

c_k: Cost of bin type k

w_{mj}: Weight of item j in m category

v_{mj}: Volume of item j in m category

W_k: Weight capacity of bin type k

V_k: Volume capacity of bin type k

l_{mr} : A parameter used to avoid joint assignment of item categories m and r which are in conflict. It equals to 1 if item category m and r can be placed into the same bin, 0 otherwise

Decision Variables

$$y_k: \begin{cases} 1 & \text{if bin type k is used} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{mjk}: \begin{cases} 1 & \text{if item j in category m is assigned to bin type k} \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Model

$$\text{Min } \sum_{k=1}^{K} y_k * c_k \tag{1}$$

$$\text{s.t. } \sum_{m=1}^{M} \sum_{j=1}^{n} w_{mj}\, x_{mjk} \leq W_k y_k \qquad k = \{1, \dots, K\} \tag{2}$$

$$\sum_{m=1}^{M} \sum_{j=1}^{n} v_{mj}\, x_{mjk} \leq V_k y_k \qquad k = \{1, \dots, K\} \tag{3}$$

$$\sum_{k=1}^{m} x_{mjk} = 1 \qquad m=\{1, \dots, M\};\ j=\{1, \dots, n\} \tag{4}$$

$$l_{mr} \geq x_{mjk} + x_{rik} - 1 \qquad \text{for all } m; r; i; j; k; m \neq r \tag{5}$$

$$x_{mjk} \in [0,1] \qquad m=\{1, \dots, M\};\ j=\{1, \dots, n\};\ k=\{1, \dots, K\} \tag{6}$$

$$y_k \in [0,1] \qquad k = \{1, \dots, K\} \tag{7}$$

This formulation includes volume capacity and item categories' conflict constraints in addition to classical VSBPP formulation of Correia et al. (2008). The objective function (1) is to minimize the cost of the bins used for packing all the items. Constraints (2) and (3) ensure that items are always assigned to bins without violating the weight and volume capacity of bins. Constraint (4) guarantees that each item is packed and assigned to only one bin. Constraint (5) ensures that when item category m and r are in conflict, the items in these categories can not be in the same bin. Constraints (6) and (7) are domain constraints.

Since the problem is NP-Hard, obtaining optimal results using exact algorithms may not be possible within a reasonable time due to those problems' size and complexity. For complex variable sized bin packing problems, approximation algorithms and many metaheuristic approaches have been used (Delorme, Iori, & Martello, 2016). From this point of view, Genetic Algorithm, one of the most successful metaheuristics, is applied in this paper.

III. SOLUTION MODEL

III.I. Genetic Representation for VSBPP

Genetic algorithms are discovered as useful tools for search an optimization problems; such algorithms are classified as stochastic since randomness has an essential role (Sivanandam, & Deepa, 2008). In genetic algorithms, the term 'chromosome' refers to a candidate solution to a problem; the chromosomes are populated, selected according to fitness, and crossed over to generate new offspring of chromosomes (Mitchell, 1998).

In a VSBPP problem, the solution is essentially an arrangement of items into bins. Since the bins have limited capacity, each arrangement must satisfy the capacity constraints. Moreover, each item appears exactly once in a valid chromosome (Falkenauer, 1996). The genetic model developed for VSBBP problems should utilize a genetic representation that can formulate any valid solution that conforms to the constraints.

Given an input that consists of *m* bins and *n* items, the genetic representation of VSBPP can be generalized as a chromosome that consists of *m* x *n* genes. Since the bin capacity can be hypothetically large enough to contain all of the items, such a genetic representation is sufficient to represent every feasible solution. Table-I exhibits a chromosome that formulates the genetic representation of a VSBBP with an input of two bins and five items.

Table I. The Genetic Representation of a VSBBP for 2 Bins & 5 Items

	Item#1		Item#4		Item#3	Item#5			Item#2
BIN #1					BIN #2				

According to the genetic representation above, a random solution would be a random distribution of items into the slots in the chromosome. Such a distribution would result in a vast number of permutations, since the value of permutation (n, m x n) is huge as the values of m and n get considerably larger. However, such permutations often represent the same solution. For instance, swapping Item #3 and Item #5 would change the genetic representation in the sample chromosome exhibited in Table I; however, such a modification in the chromosome would represent exactly the same solution. Minimization of redundancy is an important design principle for constructing genetic representations; in this regard, each solution should be represented by as few distinct chromosomes as possible (Falkenauer, 1996).

III.II. Initial Population Generation

In genetic algorithms, the initial population usually consists of valid individuals generated at random (Falkenauer & Delchambre, 1992). In VSBPP, the validity criteria consist of the problem capacity constraints defined for the bins. Moreover, as shown in problem formulation, a valid individual should contain an arrangement of all items into the bins.

The solution model involves the initial population generation at random. This step requires a random distribution of items into the bins. Each random chromosome is evaluated according to the constraints specified in the input. In our problem, the capacity constraints are separately defined in terms of volume and weight. Furthermore, the existence of item conflicts requires additional controls on the individuals. As an example, one might define a rule such as “Items of food category should not be placed in a bin that has already contained chemicals”. The model also takes account of such rules as constraints. The randomly generated valid chromosomes that satisfy all constraints are included in the initial population.

III.III. Crossover and Offspring Generation

A new generation of chromosomes is formed through crossover in each iteration. Population diversity is crucial to the genetic algorithm’s ability to continue fruitful exploration (Nsakanda, Price, Diaby, & Gravel, 2007). In this regard, selection of crossover operators is important. Various strategies are employed for crossover, such as single-point crossover, two-point crossover, multi-point crossover, uniform crossover, three parent crossover, shuffle crossover, ordered crossover, etc. (Sivanandam & Deepa, 2008). In single-point, two-point or N-point crossover strategies the new individuals are generated with chunks of genes copied from the parents. On the other hand, each gene is inherited randomly from one of the parents in uniform crossover strategy. In our study, a uniform crossover strategy was used to promote population diversity.

The objective of the crossover is to generate a number of healthy individuals in the new offspring. In our study, tests were executed with population sizes of 50, 100 and 200. Each test includes a random population of individuals and 50 consecutive crossovers. The newly bred chromosomes are checked against capacity constraints, conflict criteria, and finally checked to avoid redundancy. Before the next iteration, the ancestors and newly populated offspring is combined, and the individuals with the minimum cost are selected for the next crossover phase.

III.IV. Fitness Function

In a bin-packing problem, the quality of a solution can be intuitively expressed in terms of the number of bins used. Burke, Hyde, Kendall, & Woodward (2012) argued that using simply the number of bins lead to problem of plateaus in the search space. However, the assumptions of our problem involve that the cost of the bins varies as well as bin capacity. Since the objective of the problem is to find solutions with minimum cost, the fitness of a solution is formulated as the total cost of bins used.

III.V. Input Data and Criteria

The input consists of sample data with 50 items of seven categories. The inputs vary in terms of weight and volume. As an assumption, item category is used to define the item conflicts. The weight, volume and category of items are listed in Table II.

Table II. Weight, Volume and Category of the Items

Item No	Category	Weight (kg)	Volume (cm^3)	Item No	Category	Weight (kg)	Volume (cm^3)	Item No	Category	Weight (kg)	Volume (cm^3)
0	1	1100	500	**17**	4	440	200	**34**	5	45	1500
1	1	240	450	**18**	4	100	200	**35**	5	300	600
2	1	200	500	**19**	4	1600	400	**36**	6	100	200
3	1	300	600	**20**	4	80	125	**37**	6	115	100
4	2	1300	2000	**21**	4	160	275	**38**	6	635	100
5	2	260	500	**22**	4	1600	200	**39**	6	120	110
6	3	250	350	**23**	4	40	100	**40**	6	310	170
7	3	200	600	**24**	4	90	225	**41**	6	90	120
8	3	440	200	**25**	4	100	600	**42**	6	500	200
9	3	300	200	**26**	5	150	430	**43**	6	220	100
10	3	520	800	**27**	5	145	1000	**44**	6	145	600
11	3	230	400	**28**	5	90	160	**45**	6	600	100
12	3	600	400	**29**	5	560	750	**46**	7	300	180
13	3	300	590	**30**	5	220	1270	**47**	7	180	90
14	3	400	500	**31**	5	140	1500	**48**	7	200	225
15	3	600	300	**32**	5	180	200	**49**	7	380	400
16	4	210	200	**33**	5	120	400				

Table III exhibits the bins with variable sizes that differ in capacity and cost.

Table III. Constraints for Bins and Bin Costs

Bin Type	Available Number	Cost (units)	Maximum Volume (cm^3)	Maximum Weight (kg)
1	4	350	16000	7000
2	4	250	7000	4500
3	4	180	4800	2200
4	4	525	26330	19000

Since the VSBP problem addressed has item conflicts, a list of rules that define the co-existence of items is required. It is assumed that the items conflicts exist among item categories. The conflict constraints across item categories is exhibited in Table IV.

Table IV. Item Conflicts Defined In Terms of Categories

Category	1	2	3	4	5	6	7
1		X					
2	X						X
3							
4							
5							
6							
7		X					

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In order to test the genetic model presented, an application has been developed using C# in Microsoft Visual Studio 2015. Sample data provided in Table I, Table II, Table III were used in order to generate results. At the end of each iteration, the software provides a copy of the population as well as the duration of the calculations in a file. The application is designed to use the population size and iteration count provided as parameters. In addition, all tests are executed without the item conflicts.

Initially, the model was used to generate results with 50 items 16 bins. In the initialization and offspring generation phases, the number of individuals was supplied as 50, 100 and 200. For each of the population size, 100 tests were carried out. In addition, the items and bins were simply duplicated to perform the tests for larger scale. Resultantly, 600 tests were performed with various combinations of inputs and population sizes. Test results with conflicts are included in Table V.

Table V. Results of Tests for Various Population Sizes (With Conflicts)

	50 Items, 16 Bins, With Conflicts			**100 Items, 32 Bins, With Conflicts**		
Population Size	**50**	**100**	**200**	**50**	**100**	**200**
Min. Cost Obtained (units)	1650	1300	1050	3465	2400	2080
Min. Cost Avg. (units)	2173	1712	1404	4857	3284	2378
Avg. # of Iteration for Min. Cost	19,18	22,47	24,41	23,08	28,37	29,95
Avg. Duration / Iteration (ms)	74	144	294	379	897	2559

In order to compare the effect of conflicts on the results, all tests were performed without the conflicts. The results are given in Table VI.

Table VI. Results of Tests for Various Population Sizes (Without Conflicts)

	50 Items, 16 Bins, No Conflicts			**100 Items, 32 Bins, No Conflicts**		
Population Size	**50**	**100**	**200**	**50**	**100**	**200**
Min. Cost Obtained (units)	1550	1050	1050	2760	2330	2080
Min. Cost Avg. (units)	2078	1645	1318	4225	2870	2376
Avg. # of Iteration for Min. Cost	19,61	22,99	25,63	24,60	27,70	31,56
Avg. Duration / Iteration (ms)	72	106	230	257	674	5128

In a typical optimization problem, an increase in costs can be expected, as the constraints get tighter. Therefore, it can be argued that the tests executed without conflicts must result in average smaller costs. Indeed, the average minimum costs obtained in tests without conflicts are smaller in both input scales and all population sizes. Moreover, the tests without item conflicts mostly performed faster. Exceptionally, tests with 100 items and population size of 200 performed faster with conflicts. The subsequent tests to check the consistency of this finding evaluated in similar results. A possible explanation of this finding is that the conflicts lead to more separately arranged items where the fail possibility due to capacity constraints is lower.

In Table V and Table VI, it can be noticed that in most tests the minimum score is obtained at iterations between 20 and 30. Due to preliminary tests, each test was planned with 50 consecutive crossovers after the initialization. It can be claimed that in the tests with a higher population size, the results improve in more iterations.

In the results given in Table V and Table VI, it is noticeable that the minimum costs are obtained in the tests with largest population size. On the other hand, the time required to complete better results are significantly longer. In fact, it was predictable to notice trade-offs between minimum costs obtained in the tests and average duration of iterations. Such a relationship is exhibited in Figures I and II.

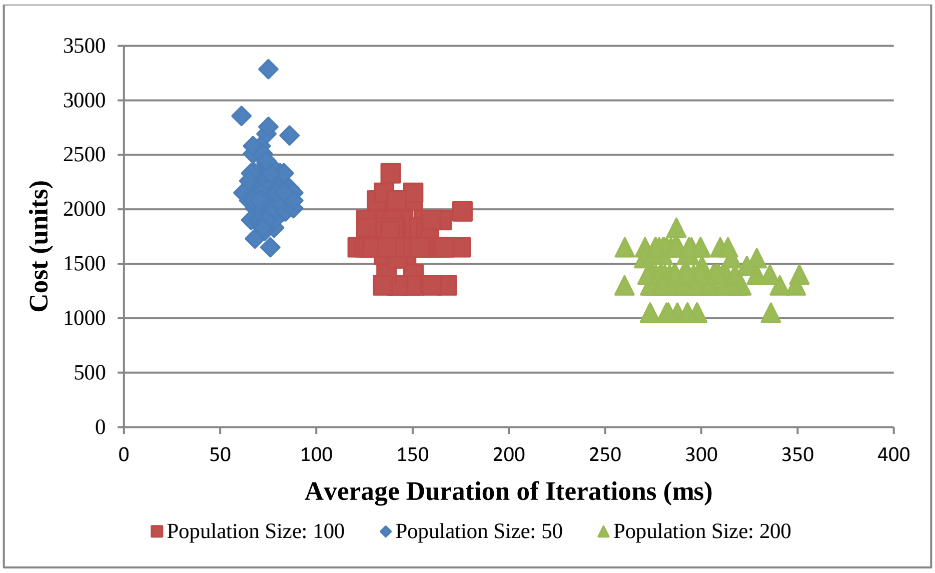

Figure I. Tradeoff between best cost with duration (50 Items, 16 Bins, With Conflicts)

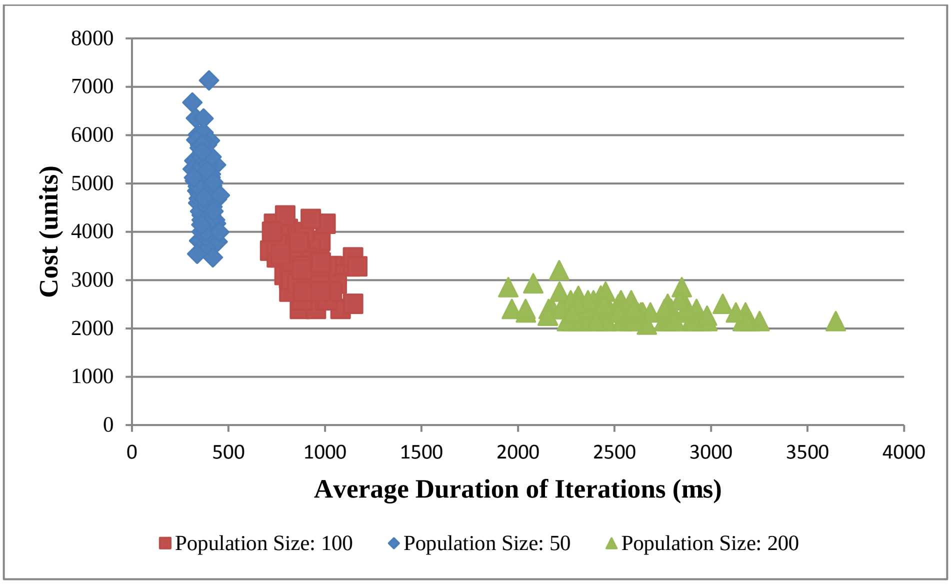

Figure II. Tradeoff between best cost with duration (100 Items, 32 Bins, With Conflicts)

Each test consists of an initialization phase and 50 offspring generated with consecutive crossovers. As each offspring is generated, it is aimed to obtain individuals with better fitness score. In our model, it can be observed that the minimum costs decrease in further crossovers.

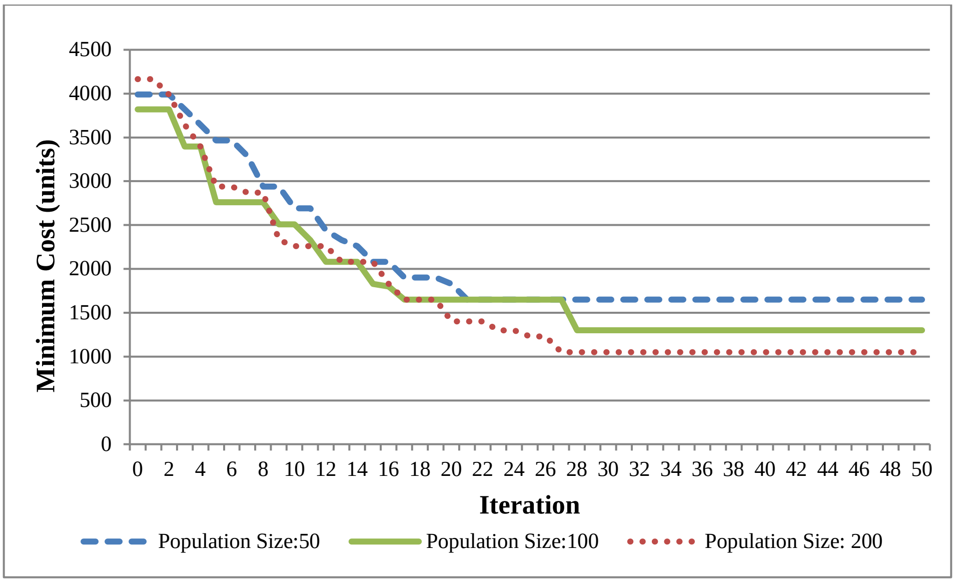

Figure III. Min. Cost by Iteration (50 Items, 16 Bins, With Conflicts)

In Figure III and Figure IV, the graph of minimum costs achieved in each iteration are depicted, from the initial population to the last offspring.

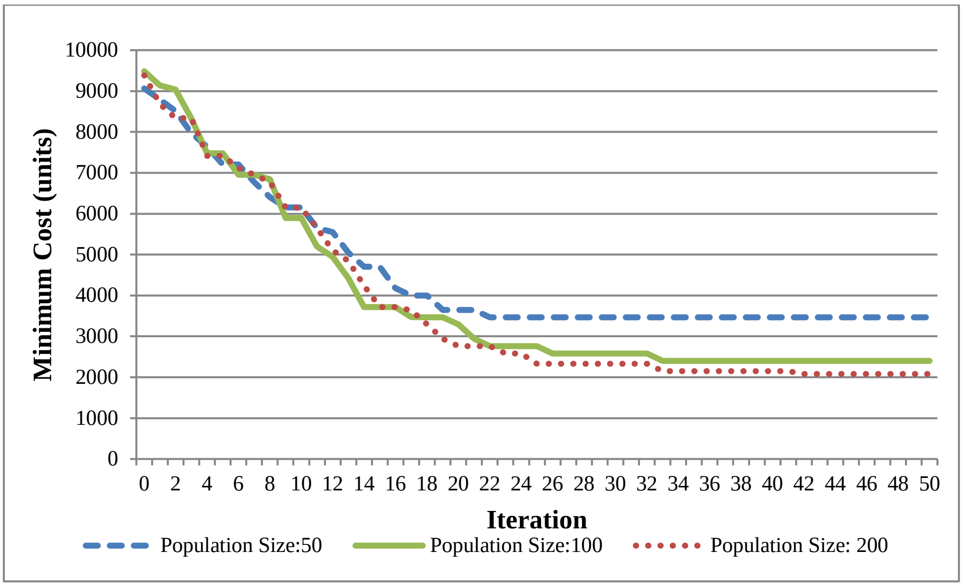

Figure IV. Min. Cost by Iteration (100 Items, 32 Bins, With Conflicts)

To demonstrate the performance of the genetic approach in our problem, each attempt to produce a valid chromosome was logged to generate statistics. The validity of an individual is checked against the capacity and conflict constraints. Moreover, chromosomes eliminated due to redundancy were also reported in Table VII.

Table VII. Statistics on the Validity of Individuals at Initialization and Crossover

Phase	Items	Population	Attempts	Failures	Conflicts	Redundant	Success	Failure Rate (%)	Conflict Rate (%)	Redundancy Rate (%)	Success Rate (%)
Initialization	50	50	3.213.624	1.822.609	944.596	441.506	4.913	56,7151	29,3935	13,7386	0.1529
Initialization	50	100	6.078.831	3.423.945	1.794.209	850.799	9.878	56,3257	29,5157	13,9961	0.1625
Initialization	50	200	10.769.452	6.069.206	3.151.620	1.528.975	19.651	56,3558	29,2644	14,1973	0.1825
Initialization	100	50	5.281.637	4.228.397	923.886	124.341	5.013	80,0585	17,4924	2,3542	0.0949
Initialization	100	100	9.602.425	7.641.973	1.712.083	238.381	9.988	79,5838	17,8297	2,4825	0.1040
Initialization	100	200	17.791.196	14.148.533	3.187.221	435.443	19.999	79,5255	17,9146	2,4475	0.1124
Crossover	50	50	295.556	10.057	25.334	10.192	249.973	3,4027	8,5716	3,4484	84.5772
Crossover	50	100	627.340	55.502	64.514	7.365	499.959	8,8472	10,2837	1,1740	79.6951
Crossover	50	200	1.374.464	221.329	149.593	3.593	999.949	16,1029	10,8837	0,2614	72.7519
Crossover	100	50	333.105	35.876	40.594	6.650	249.985	10,7702	12,1865	1,9964	75.0469
Crossover	100	100	834.197	215.145	115.790	3.282	499.980	25,7907	13,8804	0,3934	59.9355
Crossover	100	200	2.600.237	1.305.399	293.473	1.407	999.958	50,2031	11,2864	0,0541	38.4564

In Table VII, it can be noticed that the count of attempts required to generate the same number of valid individuals is dramatically lower in the crossover phase. Accordingly, the success rate of individuals is dramatically higher, while the conflict rate and failure rate is significant lower in crossovers.

Another prominent finding is that the failure rate due to capacity is higher when the input scale is larger. When the item count is scaled up by two, the complexity of the problem greatly increases due to the permutation of twice number of items into twice number of bins.

V. CONCLUSION

VSBP problems have various application areas including packing, cutting, loading and scheduling (Haouari & Serairi, 2009). In this paper, a model based on GA was proposed for a VSBP problem with item conflicts. The genetic model was tested on a sample case. Tests were executed with various input scales and population sizes. Additionally, the problem is solved without item conflicts to observe the behavior of the genetic model in both scenarios.

The findings mostly overlap with previous GA and VSBP studies. As expected, the test results indicated that iterations with crossovers generate offspring with better fitness scores than brute force approach. Mostly, the fitness scores improve over 20-30 iterations. Figures III and IV demonstrate the improvement through consecutive iterations with crossover for different population sizes.

The scalability of the model was tested with two different sets of inputs. For the first sample input that consists of 50 items and 16 bins, the best result was a minimum of 1050. As the input was duplicated, the best result obtained was 2080 resulting in a slight improvement on the cost per item. Both scores were obtained in the tests with largest population size. In general, it has been noticed that larger population sizes lead to individuals with better fitness scores. However, the decline in cost was accompanied with an increase in calculation time. The tradeoff between the time of computation and costs obtained was exhibited in discussion.

It has been noticed that the existence of conflicts in the constraints mostly increases the duration of the iterations. Besides, with a decent population size and sufficient iterations for crossover, the model helped to obtain the same results without the conflicts. Considering the limitations of the study, it can be claimed that such a finding is limited for the inputs in our case.

The genetic algorithms mostly rely on the use of selection, crossover and mutation operators (Sivanandam & Deepa, 2008). Various crossover strategies have been proposed including Simplex crossover (Tsutsui, Yamamura, & Higuchi, 1999), multi-parent partially mapped crossover (Ting, Su, & Lee, 2010). In further studies, GA solutions for VSBP problems with conflicts can be reinforced with mutation operators, and tested with adoption of different crossover operators. Furthermore, the efficiency of genetic models for VSBP problems can be enhanced with the use of heuristics.

REFERENCES

Bang-Jensen, J. & Larsen, R. 2012. Efficient algorithms for real-life instances of the variable size bin packing problem. *Computers & Operations Research*, 39(11): 2848-2857.

Belov, G. & Scheithauer, G. 2002. A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research*, 141(2): 274-294.

Bodis, A. 2015. Bin packing with directed stackability conflicts. *Acta Universitatis Sapientiae, Informatica*, 7(1): 31-57.

Burke, E. K., Hyde, M.R., Kendall, G., & Woodward, J. 2012. Automating the packing heuristic design process with genetic programming. *Evolutionary Computation*, 20(1): 63-89.

Correia, I., Gouveia, L., & Saldanha-da-Gama, F. 2008. Solving the variable size bin packing problem with discretized formulations. *Computers & Operations Research*, 35(6): 2103-2113.

Delorme, M., Iori, M., & Martello, S. 2016. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255: 1-20.

Dokeroglu, T., & Cosar, A. 2014. Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. *Computers & Industrial Engineering*, 75: 176-186.

Epstein, L., & Levin, A. 2008. On bin packing with conflicts. *SIAM Journal on Optimization*, 19(3): 1270-1298.

Epstein, L., Favrholdt, L.M., & Levin, A. 2011. Online variable-sized bin packing with conflicts. *Discrete Optimization*, 8(2): 333-343.

Falkenauer, E., & Delchambre, A. 1992. A genetic algorithm for bin packing and line balancing. In *Proceedings of IEEE International Conference on Robotics and Automation.* 1186-1192.

Falkenauer, E. 1994. New representation and operators for GAs applied to grouping problems. *Evolutionary Computation*, 2(2): 123-144.

Falkenauer, E. 1996. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2(1): 5-30.

Friesen, D.K., & Langston, M.A. 1986. Variable sized bin-packing. *SIAM Journal on Computing*, 15(1): 222-230.

Haouari, M., & Serairi, M. 2009. Heuristics for the variable sized bin-packing problem. *Computers & Operations Research*, 36(10): 2877-2884.

Haouari, M., & Serairi, M. 2011. Relaxations and exact solution of the variable sized bin packing problem. *Computational Optimization and Applications*, 48(2): 345-368.

Iima, H., & Yakawa, T. 2003. A new design of genetic algorithm for bin packing. *The Congress on Evolutionary Computation - CEC '03*, 1044-1049.

Jansen, K. 1999. An approximation scheme for bin packing with conflicts. *Journal of Combinatorial Optimization*, 3(4): 363-377

Kang, J., & Park, S. 2003. Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147(2): 365-372.

Kinnersley, N.G., & Langston, M.A. 1988. Online variable-sized bin packing. *Discrete Applied Mathematics*, 22(2): 143-148.

Korte, B., & Vygen, J. 2006. Bin-packing, combinatorial optimization: theory and algorithms. *Algorithms and Combinatorics.* 21. Springer, 426-441.

Li, K., Liu, H., Wu, Y., & Xu, X. 2014. A two-dimensional bin-packing problem with conflict penalties. *International Journal of Production Research*, 52(24): 7223-7238.

Maiza, M., Labed, A., & Radjef, M.S. 2013. Efficient algorithms for the offline variable sized bin-packing problem. *Journal of Global Optimization*, 57(3): 1025-1038.

Martello, S., Pisinger, D., & Vigo, D. 2000. The three dimensional bin packing problem. *Operations Research*, 48(2): 256-267.

Mitchell, M. 1998. *An introduction to genetic algorithms*. Springer.

Mohamadi, N. 2010. Application of genetic algorithm for the bin packing problem with a new representation scheme. *Mathematical Sciences*, 4(3): 253-266.

Monaci, M. 2003. Algorithms for packing and scheduling problems. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(1): 85-87.

Nsakanda, A.L., Price, W.L., Diaby, M., & Gravel, M. 2007. Ensuring population diversity in genetic algorithms: A technical note with application to the cell formation problem. *European Journal of Operational Research*, 178(2): 634-638.

Quiroz-Castellanos, M., Cruz-Reyes, L., Torres-Jimenez, J., Gomez S.C., Huacuja, H.J.F., & Alvim, A.C.F. 2015. A grouping genetic algorithm with controlled gene transmission for the bin-packing problem. *Computers & Operations Research*, 55: 52-64.

Reeves, C. 1996. Hybrid Genetic Algorithms for bin-packing and related problems. *Annals of Operations Research*, 63(3): 371-396.

Sadykov, R., & Vanderbeck, F. 2013. Bin packing with conflicts: a generic branch-and-price algorithm. *INFORMS Journal on Computing*, 25(2): 244-255.

Sivanandam, S.N., Deepa, S.N. 2008. *Introduction to genetic algorithms*. Springer Science & Business Media.

Ting, C.K., Su, C.H., & Lee, C.N. 2010. Multi-parent extension of partially mapped crossover for combinatorial optimization problems. *Expert Systems with Applications*, 37(3): 1879-1886.

Tsutsui, S., Yamamura, M., & Higuchi, T. 1999. Multi-parent recombination with simplex crossover in real coded genetic algorithms. *In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*. 1: 657-664. Morgan Kaufmann Publishers Inc.