Journal of Algorithms and Computation
journal homepage: http://jac.ut.ac.ir

Randomized Algorithm For 3-Set Splitting Problem and it's Markovian Model

Mahdi Heidari*[1], Ali Golshani†[2], D. Moazzami‡[3] and Ali Moeini§[4]

[1]Department of Information Technology, Ghent University-imec, IDLab, B-9052 Ghent, Belgium.
[2]Department of Algorithms and Computation, University of Tehran.
[3,4]University of Tehran, College of Engineering, Faculty of Enginering Science

ABSTRACT

In this paper we restrict every set splitting problem to the special case in which every set has just three elements. This restricted version is also NP-complete. Then, we introduce a general conversion from any set splitting problem to 3-set splitting. Then we introduce a randomize algorithm, and we use Markov chain model for run time complexity analysis of this algorithm. In the last section of this paper we introduce "Fast Split" algorithm.

ARTICLE INFO

Article history:
Received 30, June 2015
Received in revised form 18, February 2016
Accepted 30 March 2016
Available online 01, April 2016

Keyword: NP-complete problem, set splitting problem, SAT problem, Markov chain.

AMS subject Classification: 05C78.

1 Introduction

We want to restrict set splitting problem in general case to the specific version in which each set has just three elements. We know k-set splitting for $k \geq 3$ is NP-complete, [5]. In

*Email: mahdi.heydari@intec.ugent.be
†Email: ali.golshani@gmail.com
‡Email: dmoazzami@ut.ac.ir
§Email: moeini@ut.ac.ir

k-set splitting each set has just k elements, and there are some approximation algorithm for these restricted problems [1, 3, 4]. We first introduce a general conversion from any set splitting problem to 3-set splitting. Then we introduce a randomize algorithm, and we use Markov chain model for run time complexity analysis of this algoritm.In the last section of this paper we introduce “Fast Split” algorithm.

2 Basic Definitions

The set splitting problem is NP-complete, [2]. Now we describe the Set Splitting Problem as follows:

Instance: collection C of subsets of a finite set S.

Question: is there a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2 .

Informally we can say, we have some bins, and in each bin we have some balls, each ball can be in one or more bins simultaneously. Now we want to color balls with two colors; blue and red, in such a way that each bin has at least one red ball and one blue ball.

Definition 1: Set S is 2-colorable if and only if we have a proper set splitting for S.

Definition 2: Three-ball bins are those bins which have only three balls.

We start with an example: If we have $S = \{w_1, w_2, w_3, w_4\}$, and $C = \{\{w_1, w_2\}, \{w_2, w_3\}, \{w_3, w_4\}\}$, then a proper splitting for S can be S_1 and S_2 as follows:

$$S_1 = \{w_1, w_3\}, S_2 = \{w_2, w_4\}$$

That is we can color w_1 and w_3 with blue color, w_2 and w_4 with red color

3 Transforming to 3-set splitting

Now we want to restrict our problem to the condition in which each bin have exactly three balls. In our previous example all sets had just two balls Let $S = \{w_1, w_2, \ldots, w_n\}$ be a set of balls and $C = \{c_1, c_2, \ldots, c_m\}$ be a set of bins making up an arbitrary instance of set splitting in general case. We will construct a collection C' of three-ball bins on a set S' of balls (we add some new ball to the S) which is 2-colorable if and only if C is 2-colorable. We replace each individual bin C_j with collection C'_j of three-ball bins with original balls in S and some new balls in U'_j which is used only in C'_j. S' is new balls set and has also two other balls which should have different color we named this two balls R and B and to make them to have different colors, we construct C'' as follows:

$$C'' = \{\{R, B, a\}, \{R, B, b\}, \{R, B, c\}, \{a, b, c\}\}$$

C'' consist of four bins and five balls{R,B,a,b,c}. We use R, B later in our transformation, but we don't use a, b and c anywhere else. C'' is a set of three-balls bin and to have C'' 2-colorable we might color a , b and c not the same, so we force R and B to have different colors.
So we have:

$$S' = S \cup (\bigcup_{j=1}^{m} S'_j) \cup \{a, b, c, R, B\}$$

and

$$C' = C'' \cup (\bigcup_{j=1}^{m} C'_j)$$

Now we can show that how to construct S' and C' from S and C. If we group our original bins into sets based on the number of balls in each bin, first group contains bins with two balls (we haven't bins with one ball because in that condition we haven't a proper splitting at all.) and second group has bins with three balls and so on. At each stage of k stage (when k is the number of balls in one bin) we construct equivalent 3-ball bins from original k-ball bin.
$K = 2$:
For sets with two balls the transformation is as follows:
For each bin with two balls we replace that bin with two new bins. we put R and B in two new bins separately. We then put two original balls which were in the original bin in both two new bins. (As we know, we can have one ball in more than one bin at the same time, but all of them should be color identical.)
We can summarize our discussion in:

$$S'_j = \{w_1, w_2\}$$

and

$$C'_j = \{\{w_1, w_2, R\}, \{w_1, w_2, B\}$$

We know that B and R have different colors, so we can't color w_1 and w_2 with the same color. With this transformation we can be sure if we have one 2-colorable bin with just two balls its equivalent 3-ball bins are also 2-colorable and vice versa. In Fig 2, we showed this transformation.
$K = 3$:
For the set with bins which has 3 balls, we don't need any further construction, and these bins have a proper condition.
$k > 3$:
In these cases (k can be 4, 5 or more) we can construct S'_j and C'_j as follows:

$$S'_j = \{w_1, w_2, \ldots, w_k\} \cup \{q_1, q_2, \ldots, q_{k-3}\} \cup \{r_1, r_2, \ldots, r_{k-3}\}$$

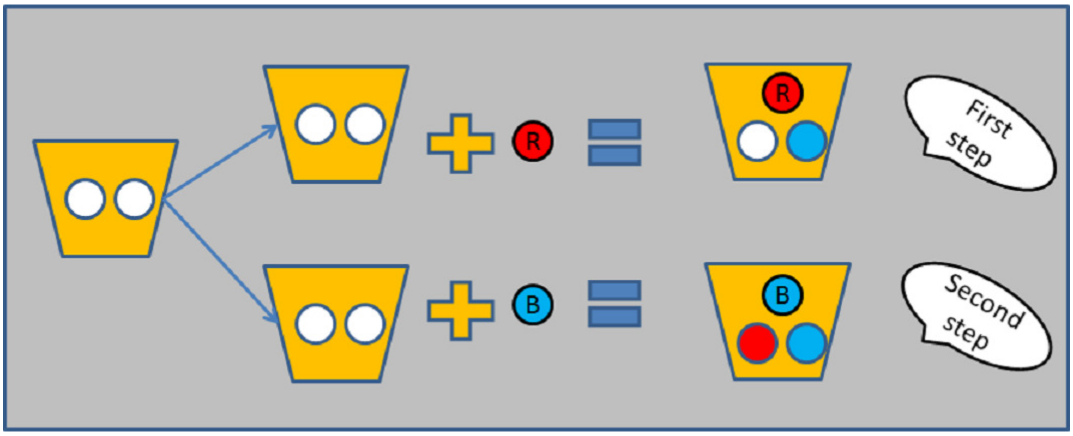

Fig.2: Transforming to 3-set splitting when k=2, in the first step we should color one of those two white balls with blue color and this make us to color the second white ball with red color in the second step.

$$C_j' = \{\{w_1, w_2, q_1\} \cup \{r_{k-3}, w_{k-1}, w_k\}\} \bigcup$$
$$\{\{r_i, w_{i+2}, q_{i+1}\}, 1 \leq i \leq k-4\} \bigcup \{\{q_i, r_i, R\}\{q_i, r_i, B\}, 1 \leq i \leq k-3\}$$

For each bin with k balls ($k > 3$) we have two groups of tree ball bins, first group have $k-2$ bins,just in the first and the last of these bins we put two original balls (w_1 and w_2 in the first bin, and w_k and w_{k-1} in the last bin) and one local ball, and in other bins of the first group we put only one original balls and two local balls. In the second group, we have $2(k-3)$ bins with 3 balls, these bins grantee q_i and r_i to have different colors, whereas we used two global balls R and B which we forced them before to have different colors, so to have 3-colorable bins in the second group q_i and r_i shouldn't have same color.
We follow this case with an example:
Suppose we have a bin with five balls;
$C_j = \{w_1, w_2, w_3, w_4, w_5\}$.
We construct an equivalent bins with three balls;

$$C_j' = \{\{w_1, w_2, q_1\}, \{r_1, w_3, q_2\}\ \{w_4, w_5, r_2\}\}$$
$$\cup\{\{q_1, r_1, R\}, \{q_1, r_1, B\}, \{q_2, r_2, R\}, \{q_2, r_2, B\}\}$$

Now our construction is completed, and it is easy to verify that, our construction is in polynomial time. We should show:

Claim 1: we can color the balls in the original bins with two colors ***if and only if*** we are able to color our new construction of bins and balls with two colors.

Proof: Suppose that $f : S \to \{Red, Blue\}$ is a right coloring for C.We will show f can be extended to a right coloring $f' : S' \to \{Red, Blue\}$ for C'.

We color R,B,a,b,c in the way that C'' members include both colors, and this implies that R and B have different colors.
Since the variables in $S' - (S \cup \{R, B, a, b, c\})$ are partitioned into sets S'_j and since the variables in each C'_j occurs only in sets belonging to S'_j, we need only show how f can be extended to the sets S'_j one at a time and in each case we need only verify that all the sets in the corresponding C'_j are 2-colorable. we can do this as follows: If we are able to color balls in the general form with two colors then we know, each bin with any number of balls which we say it k, has at least one red and one blue ball. If k is two, then one of them w_1 or w_2 is blue, and another one is red, so adding new ball with any color don't change its 2-colorability.

For $C_j = \{w_1, \ldots, w_k\}, k > 3$ since f is a right coloring for C there must be at least integer L such that w_L and w_{L+1} color is different. (whereas in the set C_j we should have both colors, so L exists in C_j), and also we define *not* function as follows:
not(red)=blue, not(blue)=red
we define coloring function f' on set s'_j as follows:
$L = 1:$

$$\begin{aligned} f'(r_i) &= not(f(w_{i+2})) \quad i = 1, \ldots, k-3 \\ f'(q_i) &= not(f'(r_i)) \end{aligned}$$

$2 \leq L \leq k-2$

$$\begin{aligned} f'(q_i) &= not(f(w_{i+1})) \quad i = 1, \ldots, L-1 \\ f'(r_i) &= not(f(w_{i+2})) \quad i = L, \ldots, k-3 \\ f'(q_i) &= not(f'(r_i)) \quad i = L, \ldots, k-3 \\ f'(r_i) &= not(f'(q_i)) \quad i = 1, \ldots, L-1 \end{aligned}$$

$L = k-1:$

$$\begin{aligned} f'(q_i) &= not(f(w_{i+1})) \quad i = 1, \ldots, k-3 \\ f'(r_i) &= not(f'(q_i)) \quad i = 1, \ldots, k-3 \end{aligned}$$

In the previous definitions we can verify following conditions:
$L = 1:$

$$\begin{aligned} color(w_1) &\neq color(w_2) \\ color(r_i) &\neq color(w_{i+2}) \quad 1 \leq i \leq k-3 \end{aligned}$$

$1 < L < k - 1:$

$$\begin{aligned} color(q_i) &\neq color(w_{i+1}) \quad 1 \leq i \leq L-1 \\ color(r_i) &\neq color(w_{i+2}) \quad L-1 \leq i \leq k-3 \end{aligned}$$

$L = k - 1:$

$$\begin{aligned} color(q_i) &\neq color(w_{i+1}) \quad 1 \leq i \leq k-3 \\ color(w_{k-1}) &\neq color(w_k) \end{aligned}$$

and these observations warranties the correctness of above coloring.
Conversely if f' is a right coloring for C' , it is easy to verify that the restriction of f' to the variables in S must be a right coloring for C.We should just show if C'_j have a right coloring then all w_i's ($i = 1, \ldots, k$) have not the same color. For k=2 it is obvious, but for $k \geq 3$ supposing all w_i's have a same color, without lose of generality we suppose all of them are blue. We do following assignment:
Red color=0
Blue color=1
Now the sum of colors in C'_j should be at least $|C'_j| = k - 2$ (because each set C'_j has at least one red color) whereas every w_i, q_i and r_i appears exactly one time in C'_j then we have:

$$\sum_{x \in C'_j} color(x) = \sum_{i=1}^{k} color(w_i) + \sum_{i=1}^{k-3} color(r_i) + \sum_{i=1}^{k-3} color(q_i)$$

$$= 0 + \sum_{i=1}^{k-3} (color(q_i) + color(r_i))$$

$$\overrightarrow{(color(q_i) \neq color(r_i))} = \sum_{i=1}^{k-3} 1 = k - 3$$

and this is a contradiction to our assumption ($k - 3 < k - 2$).

4 Randomized algorithm for 3-set splitting problem

We introduce a randomize algorithm for coloring balls in three-ball bins and then we use Markov chain as a tools to estimate the complexity of our randomize algorithm.
It is trivial way to find a proper coloring for balls by starting with an arbitrary coloring, and then we look at each of bins and we try to find the bins which all three balls in it have the same color. We select one of these balls and change its color to another color. It may

fix our problem in this bin, but it may cause some other negative or positive effects in other bins which were two-colorable or not in last step. We can summarize our discussion as follows:

two-coloring algorithm:

1.Start with an arbitrary coloring
2.Repeat up to m times, terminate if all bins have both two color ball.
a) Choose an arbitrary bin that its entire three balls have the same color.
b) Choose uniformly at random one of the balls in the bin and switch the color of that ball.
3. If all bins have 2 color balls return it.
4. Otherwise return the coloring is not doable.

We have three choices to select one ball in one bin, and our random generator function decides to select which ball to switch its color. In the algorithm, n denote the number of balls, and m is a controlling parameter to have a correct answer.

4.1 Markov chain model

Let S be one right coloring for n balls and let A_i represent the ball coloring after the i iteration of the algorithm and X_i denote number of balls in the current coloring A_i which their color is similar to the color of balls in the right coloring S. when $X_i = n$ or $X_i = 0$, (in one right coloring state if we switch the color of all balls red to blue and blue to red, we have still another right coloring). We found our desired coloring, and the algorithm halts with correct answer. In fact, algorithm could terminate before X_i reaches to n or 0 if it find another correct answer but for our analysis in the worst case, algorithm only stops when $X_i = 0$ or $X_i = n$. Starting with one arbitrary coloring when $0 < X_i < n$ we consider how X_i evolves over time, and in particular how long it takes before X_i reaches to n or 0.
For the case in which $0 < Xi < n$ we choose a bin that all its three balls have the same color, in this bin we have at least one ball with the right color and we have at least one ball with the wrong color. If $X_i = k$ then with probability $\frac{(k-1)}{(n-2)}$ the third ball has a right color. If we chose one ball to change its color, with probability p we increase the number of matches and with probability $1 - p$ we decrease the number of matches where p depends to the number of wrong colored balls in bin, if we have two wrong colored ball then the probability of moving forward(increasing the number of matches) is $\frac{2}{3}$ and if we have one wrong cohered ball, then the probability of moving forward is $\frac{1}{3}$. Hence we can find $Pr(X_{i+1} = j + 1|X_i = j)$ which is the probability of moving forward or increasing the number of matches:

$$Pr(X_{i+1} = j + 1 \mid X_i = j) = (\frac{k-1}{n-2})\frac{1}{3} + (\frac{n-k-1}{n-2})\frac{2}{3} = \frac{2n-k-3}{3n-6} \quad (1)$$

Let n be an even number(otherwise we add one extra ball to the ball set), and $m = \frac{n}{2}$. We define R_i as $|X_i - m|$, $R_i = m$ is equivalent to the final solution of our problem. According

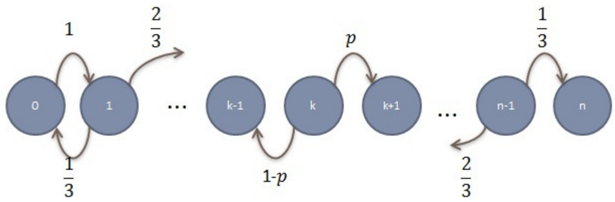

Fig.3: random walk on graph G

to (1) we have:

$$Pr(R_{i+1} = r + 1 \mid R_i = r) = \frac{1}{2} - \frac{r}{6(m-1)}) \quad 1 \leq r < m \tag{2}$$

The stochastic process $R_0, R_1, \ldots$ is a Markov chain. This chain, model a random walk on graph G as we show in Fig.3 . The vertices of G are the integers $0, \ldots, m$ and, for $1 \leq i \leq m$, node i is connected to node $i-1$ and node $i+1$.

4.2 Time complexity analysis and study of two-coloring algorithm and it's Markov chain model

Let h_r be the expected number of steps to reach m when starting from r. clearly $h_m = 0$ and $h_0 = h_1 + 1$, since from h_0 we always move to h_1 in one step. Let Z_r be random variable representing the number of steps to reach m from state r. Now consider starting from state r , where $1 \leq r \leq m-1$, with probability $1 - p_r = \frac{1}{2} + \frac{r}{6(m-1)}$ the next state is $r-1$, and in this case $Z_r = Z_{r-1} + 1$. With probability $p_r = \frac{1}{2} - \frac{r}{6(m-1)}$ the next state is $r+1$, and in this case $Z_r = Z_{r+1} + 1$. Hence

$$E[Z_r] = E[(1 - p_r)(1 + z_{r-1}) + p_r(1 + z_{r+1})] \tag{3}$$

But $E[Z_r] = h_r$ and so, by applying the linearity of expectation, we have:

$$\begin{aligned} h_r &= (1 - p_r)(1 + h_{r-1}) + p_r(1 + h_{r+1}) \\ &= (1 - p_r)h_{r-1} + p_r h_{r+1} + 1 \end{aligned}$$

We therefore have the following system of equations:

$$\begin{aligned} h_m &= 0 \\ h_0 &= h_1 + 1 \\ h_r &= (1 - p_r)h_{r-1} + p_r h_{r+1} + 1 \quad 1 \leq j \leq m \end{aligned} \tag{4}$$

By solving the above system equations we find h_0 for different values of m we reach to the following table:

m	h_0
2	6
3	15
4	32
5	58
7	160
10	586
15	4044
20	25410
30	919728
50	1053353107
75	6289502547544

If we color all balls randomly, each ball with probability $\frac{1}{2}$ have a right color, so in this case $E[X_0] = \frac{n}{2}$ and $E[R_0] = 0$. $R_i = m$ state is equivalent to the final solution of our problem, and our goal is reaching or at least getting close to this state.
Suppose Y be the probating of having both coloreds in one bin, following lemma indicates the relation between Y and the states of Markov chain.

Lemma 1: $E[Y \mid X_i = k] = 1 + p^2 - p \;\; where \;\; p = \frac{k}{n}$ and $E[Y \mid R_i = r] = \frac{3}{4} + \frac{r^2}{n^2}$.
Proof: In state k (when $X_i = k$) each ball with probability $p = \frac{k}{n}$ has a right color similar to S. Consider bin (a, b, c) and suppose (a^*, b^*, c^*) is a right coloring of this bin equivalent to S. In S, the color of two balls are different from the third one. Without lose of generality we take a^* and b^* identical and different from c^*. Now the probability of all three balls (a,b and c) have a same color is equal to:

$$\begin{aligned} Pr(a = a^*, b = b^*, c \neq c^*) + Pr(a \neq a^*, b \neq b^*, c = c^*) &= \\ p \times p \times (1-p) + p \times (1-p) \times (1-p) &= p(1-p) \end{aligned}$$

So the probability of having both colors in that bin is:

$$1 - p(1-p) = 1 + p^2 - p$$

and the second statement is verifiable according to the definition of R_i in the first statement.

The study of $E[Y]$ shows in state $R_i = m$, we expect to have maximum number of 2-colored bins, indeed in this state all bins have both colors, the minimum value of $E[Y]$ is obtained when $R_i = 0$, and in this state we have:

$$E(Y \mid R_i = 0) = \frac{3}{4}$$

If all balls colored randomly,the expectation value of R is 0, so if we start with a random coloring, we are presumably in the worst point of chain.
we define $T(k, m, p)$ for further lemma as below:

$$T(k, m, p) = \frac{1}{(1-2p)^2} \times (2p(1-p)(t^m - t^k) - (1-2p)(m-k))$$

$$0 \leq k \leq m \quad , 0 \leq p \leq 1 \;\; p \neq \frac{1}{2}, \quad t = \frac{1-p}{p}$$

Lemma 2:

$$x_m = 0$$

$$x_0 - x_1 = 1$$

$$x_k = 1 + p x_{k+1} + (1-p) x_{k-1} \quad 1 \leq k \leq m$$

the solution of above system of equations with respect of $p \neq \frac{1}{2}$ is:

$$x_k = T(k, m, p) \;\; 0 \leq k \leq m$$

you can easily prove this lemma(by substitution).

In Markov chain the probability of moving forward when we are in $R_i = r$ is equal to:

$$p_r = \frac{1}{2} - \frac{r}{6(m-1)}$$

this probability depends to r, and decreases by increasing r. If $U(i, j)$ be the expectation number of steps to reach state j when we start from state i, according to the above lemma and decreasing p_r by increasing r we can obtain following limits for $U(i, j)$:

$$T(0, j-1, p_i) \leq U(i, j)$$

$$U(0, j) \leq T(0, j, p)$$

Result 1:

$$U(0, m) \in \Omega((\frac{7}{5})^{\frac{m}{2}})$$

Proof:

$$U(0, m) \leq U(\frac{m-1}{2}, m-1)$$

$$\leq T(0, \frac{m-1}{2}, p_{\frac{m-1}{2}}) \in \theta((\frac{7}{5})^{\frac{m}{2}})$$

This result shows expectation number of steps to reach the final solution is from exponential order, however we expected this order as we know set splitting problem is NP-complete.

Result 2:

$$r = \sqrt{m-1}, U(0,r) \in O(m)$$

Proof:

$$U(0,r) \leq T(0,r,p_r)$$
$$= (\frac{9}{2}r^2 - \frac{1}{2})(1 + \frac{2}{3r-1})^r - \frac{15}{2}r^2 + \frac{1}{2}$$
$$\cong (\frac{9}{2}r^2 - \frac{1}{2})e^{\frac{2}{3}} - \frac{15}{2}r^2 \in \theta(r^2) = \theta(m)$$

This result shows the expecting number of steps to reach state $r = \sqrt{m}$ is from $O(m)$. This result is not much surprising because for this value of r, $E[Y]$ is $\frac{3}{4} + \frac{1}{4m}$.So reaching to this point of chain have not much impact to the number of 2-colorable bins.(difference between the expectation number of 2-colorable bins in two states 0 and $\sqrt{m}$ is constant number $\frac{1}{2}$).

Result 3:

$$For \quad r = \sqrt{k(m-1)Ln(m-1)}, (k > 0) \quad we\ have:$$
$$U(0,r) \in O(\frac{m^{1+\frac{2}{3}k}}{Ln(m)})$$

Proof:

$$U(0,r) \leq T(0,r,p_r)$$
$$(q = \frac{m-1}{r})$$
$$= (\frac{9(m-1)}{2kLn(m-1)} - \frac{1}{2})(1 + \frac{2}{3q-1})^{kLn(m-1)q} - 3(m-1) - \frac{9(m-1)}{2kLn(m-1)} + \frac{1}{2}$$
$$(if \quad r \to \infty)$$
$$\cong (\frac{9(m-1)}{2kLn(m-1)} - \frac{1}{2})e^{\frac{2}{3}kLn(m-1)} - 3(m-1) - \frac{9(m-1)}{2kLn(m-1)}$$
$$= (\frac{9(m-1)}{2kLn(m-1)} - \frac{1}{2})(m-1)^{(\frac{2}{3})k} - 3(m-1) - \frac{9(m-1)}{2kLn(m-1)}$$
$$\in \theta(\frac{m^{1+\frac{2}{3}k}}{Ln(m)})$$

5 Greedy algorithm for 3-set splitting problem

5.1 Transforming 3-set Splitting problem to weighted max-cut problem

Suppose 0,1 are representing blue and red colors, with this supposition for three balls a,b and c we have:

$$f(a,b,c) = \frac{1}{2}(\mid color(a) - color(b) \mid + \mid color(a) - color(c) \mid + \mid color(b) - color(c) \mid)$$

If a,b and c have the same color f(a,b,c)=0 and when they haven't the same color f(a,b,c)=1. So we can find the number of 2-colorable bins through this sigma:

$$\frac{1}{2} \sum_{(a,b,c)\in bins} f(a,b,c)$$

If we count number of simultaneous occupance of ball a and b in bin sets with n(a,b), we can summarize the above formula as below:

$$\frac{1}{2} \sum_{a,b\in balls\ set} n(a,b)\times \mid color(a) - color(b) \mid$$

Considering complete and weighted graph G=(V,E), V is equal to balls set and the weight of each edge is equal to n(a,b), in this graph ball coloring is equivalent to partitioning the vertices into two disjoint set R and B. The sum of edges weight in cut (R,B) is equal to:

$$\sum_{a\in R.b\in B} n(a,b) = \sum_{a,b} n(a,b)\times \mid color(a) - color(b) \mid$$

In other words, the weight of cut (R,B) is two times of 2-colorable bins number. So we can transform 3-set splitting problem to weighted Max-cut problem.

5.2 Greedy algorithm with factor .75:

we consider π be an instance of 3-set splitting problem, and G=(V,E) be its equivalent weighted max-cut problem π'. We show all bins number with M and 2-colorable bins number with P. So we have:

$$M = \frac{1}{3}\sum_{a,b} n(a,b)$$

$$P = \frac{1}{2}\sum_{a,b} n(a,b)\times \mid color(a) - color(b) \mid$$

following algorithm which we call it Fast Split is a solution with factor 0.5 for π' problem. Given a graph G=(V,E) start with an arbitrary partitioning of V, we move a vertex from one side to the other if it improves the solution until no such vertex exists. The number of iteration is bounded by $O(\sum_{a,b} n(a,b)) = O(M)$, because the algorithm improves the cut value by at least 1 at each step, and the maximum cut is at most $\sum_{a,b} n(a,b)$. When the algorithm terminates, each vertex $v \in V$ has at least half of its edges in the cut(otherwise moving v to the other subset improves the solution). So the cut is at least $\sum_{a,b} n(a,b)$. If p^* be the number of 2-colorable bins in optimal solution we have:

$$p_A = \frac{1}{2} \sum_{a \in R_A, b \in B_A} n(a,b) \geq \frac{1}{2} \times .5 \sum_{a,b} n(a,b) = \frac{3}{4} M \geq \frac{3}{4} p^*$$

So the above algorithm guarantees at least $\frac{3}{4}$ of bins have both colors. we can represent this algorithm for problem π as below:

Fast Split algorithm:
Start with an arbitrary coloring
repeat
change the color of that ball which has most increment in the number of 2-colorable bins
until we haven't any increment in the number of 2-colorable bins

This algorithm is actually a local search in states space, and in each step of algorithm we have at least one increment to the number of 2-colorable bins, so the algorithm repeats at most M times. Each step also needs O(M) time to search for the ball which has most increment to the number of 2-colorable bins(by one time movement across all the balls (M balls) and changing their color, we count number of 2-colorable bins and finally we choose ball with maximum increment), so run time complexity of Fast Split algorithm is from $O(M^2)$.

Lemma 3: Fast Split algorithm guarantees at least $\frac{3}{4}$ of bins are 2-colorable.

6 Conclusion and further work:

We modeled set splitting problem with ball and bin which we want to color balls in bins with two colors in such a way that we have both two color in each bin. Then we restrict our attention to the case our bins have just 3 balls. This case is also NP-complete, and we call this condition three-ball bin. We propose a way to convert any set splitting problem to three-ball bin model. We design a randomized algorithm for this restricted problem, and finally we estimate its complexity by modeling it by Markov chain. From now any further work on set splitting can be done on only three-ball bin model because we can reach to it from any general problem in polynomial time.
We implement this algorithm by Matlab programming language, complete code of this

implementation is available as request for readers. Sat is another NP-complete problem which has some similarity with set splitting problem. Again similar to 3-set splitting, 3-sat is a special case of SAT problem. In 3-SAT each clause has only three literals. There are many ideas for solving 3-sat problem, we can look at them and reach more new innovation for solving set splitting problem.

7 Acknowledgement

This work was supported by Tehran University. Our special thanks go to the University of Tehran, College of Engineering and Department of Engineering Science for providing all the necessary facilities available to us for successfully conducting this research.

References

[1] G.Anderson, L.engebretsen, Better approximation algorithm for set splitting and not-all-equal sat,Information Processing Letters 65 (1998) 305-311 .

[2] M.Garey and D.Johnson, Computers and Intractability: A guide to the Theory of NP-completeness.

[3] . M.X. Goemans, D.P Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM 42 (1995) 1115-l 145.

[4] .V.Kann, J. Lagergren, A. Panconesi, Approximability of maximum splitting of k-sets and some other Apx-complete problems, Inform. Process. Len 58 (1996) 105-110.

[5] . L. Lovsz, Coverings and colorings of hypergraphs Proc. 4th Southeastern Conf. on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, Winnipeg (1973), pp. 312