Budgets Balancing Algorithms for the Projects Assignment

Mahdi Jemmali
Department of Computer Science and Information
College of Science in Zulfi, Majmaah University
Majmaah, 11952, Saudi Arabia

***Abstract*—This paper focused on the resolution of the project's assignment problem. Several heuristics have been developed and proposed in this paper to serve as lower bounds to our studied problem. In a developing country, it is interesting to make an equitable distribution of projects in different cities in order to guarantee equality and regional development. Each project is characterized by its budget. The problem is to find an appropriate schedule to assign all projects to all cities. This appropriate schedule seeking the maximization of the budget in the city that having the minimum budget. In this paper, six heuristics were proposed to carry out the objective of resolving the studied problem. The experimental results show that the algorithm given by the heuristic P_6^r outperforms all other heuristics cited in this paper.**

***Keywords*—Heuristic; scheduling algorithms; project assignment**

I. Introduction

The regional development is a very important thing for each country. Appropriate distribution of resources is primordial to guarantee social and economic stability. The economic cycle may directly be impacted by the development of cities. The principal main of this research is the allocation of the project. The problem is based on the fact that we have several regions and several projects to distribute to the regions. The principal goal of this work is to maximize the budget for the city that having the minimum budget. This goal will be presented as a maximization of the minimum total budget problem. This problem can be solved approximately using several heuristics. Given a set of projects to be distributed to different cities (regions). Each project has its budget and will be entirely assigned to a chosen city. Seeking the balancing of the budgets to all regions, we apply some heuristics to schedule projects on the regions to maximize the minimal total budget [1]. In our work, it is assuming that all regions are identical and have the same economic characteristic, this is meaning all regions can attract all given projects. Heuristics will be used to give a feasible dispatching solution of all projects. The unequal distribution of income across regions is studied in [2] which authors presented in the research work the theory of optimal control.

Other researchers use the operations research field applied in the allocation of investment using the dynamic programming problems with linear form. The regional allocation problem was studied to found the optimal solution proprieties [3].

In our paper we propose to make dispatching of projects on cities by using several heuristics of the follows problem. Minimizing the maximum total completion time for parallel machine $P||C_{min}$ as defined in [1]. For the latter problem, an optimal method using a branch-and-bound algorithm was given in [1]. The developed algorithm used essentially the proposed tight lower and upper bounds.

In [4], the authors developed a method for multi-mode resource with cost availability problems in the project assignment. Other approach is proposed based on differential search (DS) related to the conditions and constraints of the resource project assignment problem [5]. Recently, the author in [6] proposed three approximate Solutions to solve the problem that treated the project revenues but no experimental result is given in the latter work. Some randomized variants were developed as heuristics to solve approximately the problem of the aircraft turbines engines in [7]. Indeed, authors in the latter work show that the solution of the problem can be given by solving the maximization of the minimum completion time. This problem is the same one applied in the studied case in this paper. On other work type, authors in [8], reveal the problem of allocation resources by resolving the problem using a near-optimal solution. Recently, an equity algorithm is applied on routers to ensure the balancing assignment of data through different routers [9]. This latter research work utilizes the same approach of our studied problem. This presented paper is organized as follows. In Section 2, we present the studied problem and we give some details for the problem in general. Section 3, presents six developed lower bounds through several heuristics for the studied problem. The experimental study is done in Section 4.

II. Problem Description

The problem proposed in this paper is the maximization of the minimum total budget. This problem is described as follows. Let a set Ps that determined by the n_p projects. These projects must be distributed to a fixed number of regions n_r. We index each region R_i by the index i. The estimated budget of each project j is denoted by b_j. We can't assign the same project to different regions. Each project will be assign at only one region at a time. The studied problem concerns the maximization of the minimum total budget given by R_{min}. The cumulative budget of the project j is given by the variable Cb_j.

Example 1: Assume that the number of projects is 5 ($n_p = 5$) and the number of regions is 2 ($n_r = 2$). The budget b_j for each project j is depicted in Table I.

TABLE I. BUDGET-PROJECTS VALUES.

j	1	2	3	4	5
b_j	7	2	9	3	12

We schedule all projects on the regions applying a given algorithm. The schedule gave a result which is depicted in Fig. 1. From the latter figure, we see that region 1 has projects 3, 1 and 4. However, for region 2, projects 5 and 2 are assigned.

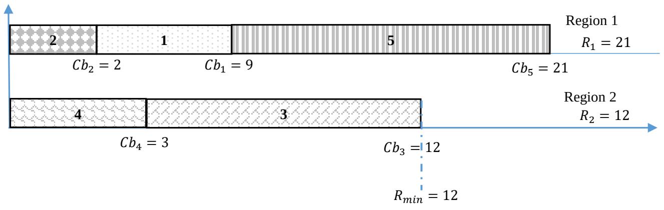

Fig. 1. Project-regions assignment.

From Fig. 1, we see that the region 1 has a total budget of 21. Nonetheless, region 2 has a total budget of 12. For this example, the maximum assigned budget is 21. However, the minimum total assigned budget is $R_{min} = 12$. The goal is to give a permutation (assignment) that maximizes the minimum of the assigned total budget of R_{min}. So, we have to search for other more performed heuristics that offers a minimum total budget better than 12.

Using the three-field notation in [10], the studied problem can be denoted as $P||C_{min}$.

III. LOWER BOUNDS

In this research work, we give a comparison of the performance of the proposed lower bounds for the studied problem with the well-known dispatching rule LPT given in [1]. Indeed, we develop six lower bounds. Two strategies are proposed in this section to develop the six algorithms. The first strategy is based on the probabilistic method applied to the choice of the project which will be assigned to the regions. The second strategy is based on the Multi-Fit algorithm to be used for the proposed problem.

A. Non-increasing Budgets Order Heuristic (NIB)

In this dispatching rule, all projects will be ordered in the non-increasing order of its budgets. After that, we assign the project having the greatest budget on the region that has the minimum total budget at this time.

B. Probabilistic based Heuristics (P_k^r)

For this type of heuristic, our study is essentially based on the selection of the project having the biggest budget to be assigned to the region that has the minimum total budget with a probabilistic approach. The probabilistic selection between the k biggest project with $k \in \{2, 3, \cdots, 6\}$ for the heuristics $\{P_2^r, \cdots, P_6^r\}$, respectively.

The chosen project is selected among the k first project having the biggest budget with probability α. We denote by $RD(a, b)$ the randomly function that return an integer between two limits (a, b). The calculation of the probabilistic selection is fixed as follows:

- $r = RD(1, k)$. The picked project will be the r^{th} biggest non-assigned project. We assign the chosen project to the region that have a minimum total budget.
- Denoted by U_p the number of non-assigned projects. If $U_p < k$, then $r = RD(1, U_p)$.

For a fixed integer k, we choose the iteration number as $limit = 1500$. The heuristic of the probabilistic project-region is the following:

Algorithm 1 Probabilistic project-region heuristic: $PPR(k)$

```
1: Set $itr = 1$.
2: Set $P_k = P$.
3: $r = RD(1, k)$.
4: Assign the $r^{th}$ biggest project ( $L_p$), to the region that
   have the minimum total budget.
5: $P_k = P_k \setminus L_p$, if $P_k \neq \emptyset$ goto 3.
6: Determine the $R_{min}^{itr}$
7: $itr = itr + +$.
8: if $itr \leq limit$ goto 2.
9: Return $PPR(k) = \max_{1 \leq itr \leq limit} R_{min}^{itr}.$ , Stop.
```

The above algorithm is given for a fixed k.

The Algorithm 1 is used when the k value is fixed in advance. To develop the algorithm of the proposed heuristic we must iterate the above as the number between 2 and k. Indeed, the final algorithm which represents the probabilistic based heuristics P_k^r needs the iteration of $PPR(j)$ from 2 to k. The calculation of the value of the heuristic P_k^r for a predetermined k is elaborated by as following algorithm.

Algorithm 2 Probabilistic algorithm P_k^r

```
for $t = 2$ to $k$ do.
    Determine $PPR(t)$.
end for
$P_k^r = \max_{2 \leq t \leq k} PPR(t)$.
```

C. Multi-fit Projects based Heuristic (MFP)

The Multi-Fit heuristic is developed in several works in literature. For our proposed research work we propose an adaptive Multi-Fit heuristic. The main idea is based to found the minimum total budget when all projects will be assigned to the given regions.

The heuristic is based on the utilization of the bin-packing approach. For our case study, the region is considered as bin and the project as an item. Once the bin capacity is fixed, the First Fit Decreasing (FFD) algorithm is used to fit projects to the bin. The first step of the algorithm is to order projects according to their budget such that $b_1 \geq b_2 \geq \ldots b_{n_p}$.

Let $L = max(b_1, b_{n_r} + b_{n_r+1}, \lceil \frac{\sum_{p=1}^{n_p} b_p}{n_r} \rceil)$ and U is the value given by applying the LPT heuristic for the problem

$P||C_{max}$. The number of reputation ite is fixed in advance. This number is used for the iteration of FFD. We set $ite = 135$ and n_b denoted the number of bins used when we apply the FFD routine.

The algorithm below describe all instructions to calculate MFP.

Algorithm 3 Multi-fit Projects algorithm (MFP)

```
1: Put $k = 0$, $u = U$ ; $l = L$.
2: Calculate $mi = \lfloor \frac{u+l}{2} \rfloor$, set $k = k + 1$.
3: Apply $FFD$ with capacity $mi$.
4: If we can assign all projects $n_p$ into $n_r$ regions, after put
   $u = mi$ and go to 5, else set $l = mi$ and go to 5.
5: If $k = ite$ then STOP, otherwise go to 2.
6: If $n_b > n_r$ then the permutation obtained by $NIB$ is
   selected, otherwise the permutation obtained by $FFD$ is
   selected. The selected permutation noted by $\sigma$.
7: Calculate the $C_{min}$ of the permutation $\sigma$.
8: $MFP = C_{min}$.
```

Example 2: Let $n_p = 10$ and $n_r = 2$. The following table represent the budget of each project.

TABLE II. Instance of budgets-projects for heuristic MFP

j	1	2	3	4	5	6	7	8	9	10
b_j	81	30	33	66	53	62	66	59	96	55

Applying the NIB heuristic for the instances given in Table II, the result will as follows. For the first region, we assign the projects $\{2, 4, 5, 8, 9\}$ with a total budget of $U = 304$. For the second region, we assign the projects $\{1, 3, 6, 7, 10\}$ with total budget $L = 297$. Then, $mi = \lfloor \frac{304+297}{2} \rfloor = 302$. Applying FFD function with capacity 302 and the teen items which is corresponding to projects in Table II. The region 1 has the projects $\{1, 4, 8, 9\}$ and the second region will have the projects $\{2, 3, 5, 6, 7, 10\}$. The first region has a total budget of 302 however the second one has a total budget of 299. Thus, $C_{min} = 299$. On the other hand, the value returned by the heuristic LPT (NIB) is 297. So, the result obtained by MFP is better than NIB.

IV. Experimental Results

We adopt several instances of classes to discuss with results and show the developed algorithms comparing with the LPT cited in previous research works.

All algorithms proposed in this paper are coded with Microsoft Visual C++ (Version 2013). The developed algorithms were coded and running on an Intel(R) Core(TM) i5-3337U CPU @ 1.8 GHz. The operating system is windows 10.
The developed algorithms were running on five types of sets of instances. These types of instances were generated as presented in [1]. We generate the budget b_j using to different distributions. Each distribution represents a class. The distribution of the classes are as following:

- Class 1: b_j is in $U[30, 100]$.
- Class 2: b_j is in $U[50, 300]$.
- Class 3: b_j is in $U[200, 500]$.
- Class 4: b_j is in $N[50, 150]$.
- Class 5: b_j is in $N[25, 500]$.

with $U[a, b]$ represent the uniform distribution between $[a, b]$ and $N[a, b]$ represent the normal distribution. The generated instances was obtained by the choice of n_p, n_r and $Class$. The pair (n_p, n_r) can have several affectation values. Indeed, we choose the affectation values given in Table III below.

TABLE III. The n_p and n_r distribution

n_p	n_r
10	2,3,5
25,50	2,3,5,10,15
100,250	3,5,10,15,25,30
300,500	10,15,30,50

Table III shows that we have in total of 1650 instances. Several indicators are given in this paper to show the performance of the proposed algorithms compared with the LPT values proposed in the previous work and presented in [1]. The indicators are given as follows.

- Lm the best heuristic value returned after running of algorithms.
- L the discussed lower bound.
- $GAP = \frac{Lm-L}{L} \times 100$.
- $Time$ the running time execution for heuristic. This time will be in seconds.
- $Perc$ the percentage among the (1650) instances that the condition $Lm = L$ is satisfied.

Table IV present the overall percentage and average time for all proposed algorithms.

TABLE IV. Overall percentage and average time comparison

	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
$Perc$	16.2%	16.1%	41.6%	52.2%	58.5%	68.2%	84.3%
$Time$	0.000	0.023	0.047	0.095	0.139	0.192	0.237

The above table shows that the heuristic that conducts the best value is P_6^r with $Perc = 84.3\%$ and $Time = 0.237$s, compared to LPT rule that the corresponding percentage is 16.2%.

The following table represents the behavior of Gap according to n_p.

TABLE V. Variation of GAP according to n_p

n_p	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
10	1.67	2.17	0.47	0.08	0.03	0.00	0.00
25	1.78	4.60	0.77	0.46	0.33	0.18	0.08
50	0.84	7.19	0.24	0.11	0.07	0.04	0.03
100	0.66	30.24	0.38	0.27	0.18	0.10	0.05
250	0.15	5.12	0.10	0.07	0.05	0.03	0.02
300	0.22	1.88	0.12	0.08	0.05	0.03	0.01
500	0.08	2.35	0.05	0.04	0.03	0.02	0.01

Table V shows that when varying the number of projects (n_p), the performance of the developed algorithms varies. The above table shows that there aren't any depends on the number of projects and the calculated GAP. For heuristics [LPT, MFP, P_2^r, P_3^r, P_4^r, P_5^r and P_6^r] the best GAP value was obtained for the following values of n_p [500, 300,500, 500, (10,500), 10, 10], respectively. Also this table shows that algorithm P_6^r having the best GAP when $n_p = 10$ and the rule LPT having the worst GAP when $n_p = 500$.

The execution time for heuristics in Table V is given in Table VI.

TABLE VI. $Time$ FOR EACH HEURISTIC AND EACH CORRESPONDING n_p

n_p	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
10	0.000	0.002	0.001	0.002	0.004	0.005	0.004
25	0.000	0.004	0.003	0.007	0.009	0.012	0.015
50	0.000	0.006	0.006	0.012	0.017	0.032	0.027
100	0.000	0.010	0.017	0.034	0.052	0.070	0.095
250	0.000	0.025	0.058	0.120	0.173	0.230	0.291
300	0.000	0.060	0.086	0.171	0.255	0.346	0.437
500	0.000	0.062	0.174	0.354	0.522	0.727	0.886

Table VII display the results of the GAP value according to the number of regions n_r. The worst GAP value is given for algorithm MFP when $n_r = 30$ and the best GAP value is given for algorithm P_6^r when $n_r = 3$ or $n_r = 50$.

TABLE VII. VARIATION OF THE GAP VALUES ACCORDING TO n_r

n_r	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
2	1.01	0.40	0.20	0.14	0.11	0.09	0.08
3	1.52	1.28	0.59	0.19	0.09	0.04	0.01
5	0.39	0.90	0.05	0.03	0.03	0.02	0.02
10	0.97	3.29	0.51	0.30	0.21	0.11	0.03
15	0.42	7.22	0.25	0.20	0.15	0.10	0.05
25	0.80	2.70	0.41	0.25	0.15	0.07	0.03
30	0.28	51.14	0.15	0.09	0.05	0.03	0.02
50	0.27	4.72	0.16	0.12	0.07	0.04	0.01

The execution time for heuristics in Table VII is given in Table VIII.

TABLE VIII. $Time$ FOR EACH HEURISTIC AND EACH CORRESPONDING n_r

n_r	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
2	0.000	0.003	0.004	0.005	0.007	0.013	0.013
3	0.000	0.001	0.013	0.025	0.037	0.054	0.070
5	0.000	0.016	0.014	0.029	0.041	0.054	0.067
10	0.000	0.040	0.050	0.101	0.149	0.202	0.251
15	0.000	0.014	0.055	0.109	0.163	0.223	0.278
25	0.000	0.033	0.045	0.093	0.135	0.183	0.230
30	0.000	0.022	0.092	0.186	0.279	0.400	0.474
50	0.000	0.088	0.154	0.322	0.465	0.619	0.790

The Table IX present the variation of the calculated GAP according to $Class$. It is clear to see that the class instances and the GAP values are not correlated.

TABLE IX. VARIATION OF THE GAP VALUES ACCORDING TO $Class$

$Class$	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
1	0.69	8.60	0.26	0.13	0.09	0.04	0.02
2	0.70	5.71	0.22	0.11	0.07	0.04	0.02
3	0.72	9.88	0.34	0.17	0.11	0.06	0.03
4	1.17	16.23	0.45	0.24	0.16	0.08	0.02
5	0.38	4.21	0.26	0.19	0.15	0.10	0.07

The execution time for heuristics and Table IX is given in Table X.

TABLE X. $Time$ FOR EACH HEURISTIC AND EACH CORRESPONDING $Class$

$Class$	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
1	0.000	0.027	0.046	0.091	0.136	0.190	0.236
2	0.000	0.021	0.048	0.094	0.140	0.205	0.242
3	0.000	0.028	0.047	0.098	0.140	0.192	0.238
4	0.000	0.003	0.047	0.096	0.141	0.188	0.234
5	0.000	0.033	0.046	0.093	0.140	0.186	0.237

For more statistics about the gap values, we present Table XI. From the Table XI, we can see that the maximum GAP is 171.59 obtained when $n_p = 100$ and $n_r = 30$ for the heuristic MFP.

TABLE XI. DETAILS FOR ALL ALGORITHMS

n_p	n_r	LPT	MFP	P_2^r	P_3^r	P_4^r	P_5^r	P_6^r
10	2	1.08	0.59	0.03	0.00	0.00	0.00	0.00
	3	3.92	3.14	1.39	0.24	0.08	0.01	0.00
	5	0.03	2.79	0.00	0.00	0.00	0.00	0.00
25	2	1.91	0.57	0.56	0.41	0.34	0.27	0.24
	3	1.64	1.94	0.73	0.26	0.11	0.04	0.00
	5	1.30	1.11	0.13	0.08	0.11	0.07	0.06
	10	4.05	16.87	2.46	1.53	1.07	0.54	0.10
	15	0.00	2.52	0.00	0.00	0.00	0.00	0.00
50	2	0.06	0.03	0.00	0.00	0.00	0.00	0.00
	3	1.57	0.73	0.49	0.21	0.10	0.05	0.01
	5	0.41	0.39	0.06	0.02	0.02	0.02	0.03
	10	1.04	1.81	0.28	0.12	0.08	0.08	0.06
	15	1.11	32.99	0.40	0.20	0.14	0.07	0.07
100	3	0.34	0.42	0.26	0.16	0.08	0.04	0.02
	5	0.16	0.18	0.06	0.03	0.02	0.01	0.01
	10	0.48	0.54	0.20	0.11	0.08	0.05	0.02
	15	1.05	4.10	0.83	0.78	0.62	0.39	0.16
	25	1.27	4.62	0.62	0.38	0.23	0.11	0.04
	30	0.68	171.59	0.32	0.16	0.07	0.03	0.03
250	3	0.11	0.17	0.10	0.08	0.05	0.03	0.01
	5	0.04	0.04	0.01	0.01	0.00	0.00	0.00
	10	0.12	0.23	0.05	0.02	0.01	0.01	0.00
	15	0.21	1.47	0.17	0.16	0.13	0.09	0.06
	25	0.32	0.78	0.20	0.12	0.08	0.04	0.01
	30	0.10	28.02	0.05	0.03	0.03	0.01	0.00
300	10	0.09	0.17	0.04	0.02	0.01	0.01	0.00
	15	0.14	0.22	0.08	0.05	0.04	0.03	0.00
	30	0.28	1.69	0.18	0.11	0.08	0.03	0.01
	50	0.36	5.43	0.20	0.15	0.08	0.05	0.02
500	10	0.05	0.10	0.03	0.02	0.01	0.01	0.00
	15	0.02	2.00	0.01	0.00	0.00	0.00	0.00
	30	0.06	3.27	0.05	0.05	0.04	0.04	0.04
	50	0.19	4.02	0.13	0.09	0.06	0.03	0.01

V. CONCLUSION

This paper treats the problem of the distribution of different projects having several budgets. The problem is NP-hard. We proposed in this paper several heuristics to give approximates solutions for the studied problem. The algorithms given in this work are tested through some classes of instances generated uniformly based on classes already proposed in the literature review. The results show that the probabilistic and iterative method outperforms the Multi-fit and dispatching rule algorithms. The results show that the algorithm P_6^r has the best GAP value through the tested instances. Future work can be done regarding the performance of other heuristics and the exact solution of the studied problem.

Acknowledgment

The authors would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project Number No. R-1441-30.

References

[1] M. Haouari and M. Jemmali, “Maximizing the minimum completion time on parallel machines,” *4OR*, vol. 6, no. 4, pp. 375–392, 2008.

[2] S. Alexiadis, “Regional allocation of investment: Implications for regional policy,” *Procedia Economics and Finance*, vol. 9, pp. 176–180, 2014.

[3] Y. Ohtsuki, “Regional allocation of public investment in an n-region economy,” *Journal of Regional Science*, vol. 11, no. 2, pp. 225–233, 1971.

[4] J.-J. Qi, Y.-J. Liu, H.-T. Lei, and B. Guo, “Solving the multi-mode resource availability cost problem in project scheduling based on modified particle swarm optimization,” *Arabian Journal for Science and Engineering*, vol. 39, no. 6, pp. 5279–5288, 2014.

[5] N. Rahmani, V. Zeighami, and R. Akbari, “A study on the performance of differential search algorithm for single mode resource constrained project scheduling problem,” *Decision Science Letters*, vol. 4, no. 4, pp. 537–550, 2015.

[6] M. Jemmali, “Approximate solutions for the projects revenues assignment problem,” *Communications in Mathematics and Applications*, vol. 10, no. 3, 2019.

[7] M. Jemmali, L. K. B. Melhim, and M. Alharbi, “Randomized-variants lower bounds for gas turbines aircraft engines,” in *World Congress on Global Optimization*. Springer, 2019, pp. 949–956.

[8] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, “Near optimal online algorithms and fast approximation algorithms for resource allocation problems,” *Journal of the ACM (JACM)*, vol. 66, no. 1, p. 7, 2019.

[9] M. Jemmali and H. Alquhayz, “Equity data distribution algorithms on identical routers,” in *International Conference on Innovative Computing and Communications*. Springer, 2020, pp. 297–305.

[10] E. L. Lawler, J. K. Lenstra, A. H. R. Kan, and D. B. Shmoys, “Sequencing and scheduling: Algorithms and complexity,” *Handbooks in operations research and management science*, vol. 4, pp. 445–522, 1993.