SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Jurnal & Penelitian Teknik Informatika
Volume 1 Nomor 2, April 2017
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Analisis Komparasi *Genetic Algorithm* dan *Firefly Algorithm* pada Permasalahan *Bin Packing Problem*

Adidtya Perdana
Sekolah Tinggi Teknik Harapan Medan
Jl. H.M. Jhoni No. 70 C Medan
adid.dana@gmail.com

Abstrak — *Bin Packing Problem* merupakan permasalahan kombinatorial yang menggambarkan bagaimana sekumpulan objek atau benda dengan ukuran yang berbeda-beda dikemas kedalam suatu wadah dengan ukuran yang telah ditentukan. Dengan ukuran yang berbeda-beda tingkat kesulitan yang dihadapi adalah bagaimana mengoptimalkan jumlah pemakaian wadah untuk mengkemas objek-objek tersebut. Dengan menggunakan metode *Heuristic* yaitu *Genetic Algorithm* dan *Firefly Algorithm* diharapkan mampu menyelesaikan permasalahan *Bin Packing Problem*. *Genetic Algorithm* merupakan metode yang diadaptasi dari genetika dan evolusi alami yang didalamnya terdapat parameter-parameter bawaan dari genetika alami. Sedangkan *Firefly Algorithm* merupakan metode yang diadaptasi dari perilaku kedipan cahaya kunang-kunang pada malam hari. Didalamnya terdapat aturan-aturan yang menyatakan kunang-kunang akan mendekati satu atau lebih cahaya yang paling terang. Pada tulisan ini, penulis akan membahas mengenai komparasi hasil yang diberikan oleh *Genetic Algorithm* dan *Firefly Algorithm* dalam menyelesaikan permasalahan *Bin Packing Problem* satu dimensi. Dengan tulisan ini diharapkan dapat memberikan sumbangan pengetahuan tentang penggunaan metode-metode heuristic dalam menyelesaikan permasalahan kombinatorial khususnya permasalahan *NP-Problem.*

Kata Kunci — *bin packing problem, genetic algorithm, firefly algorithm, np-problem, metode heuristic, permasalahan kombinatorial.*

I. PENDAHULUAN

Bin Packing Problem merupakan permasalahan kombinatorial yang termasuk kedalam *NP-Complete* (*non-deterministic polynomial complete*) yang menggambarkan bagaimana sekumpulan objek atau benda yang memiliki ukuran yang tidak sama (berbeda-beda) yang akan dipaket atau dibungkus kedalam satu wadah atau bin yang memiliki ukuran yang tetap. *Bin Packing Problem* merupakan pengembangan dari Knapsack Problem [1]. Tingkat kesulitan yang dihadapi pada permasalahan ini adalah bagaimana cara mengoptimalkan jumlah wadah atau *bin* untuk menampung seluruh objek atau benda yang diberikan. Semakin sedikit wadah atau bin yang terpakai maka semakin optimal. Dan jika semakin banyak wadah atau *bin* yang terpakai maka semakin buruk hasil penyelesaiannya.

Jika *Bin Packing Problem* yang merupakan permasalahan *NP-Problem* di reduksi menjadi *P* (*Polynomial*) *Problem* dapat diselesaikan dengan mudah menggunakan teknik *Lower Bound*. *Lower Bound* adalah indeks nilai terkecil yang digunakan untuk menghitung kompleksitas dari *Bin Packing Problem* [1][10]. Untuk menggunakan teknik *Lower Bound* pada *Bin Packing Problem*, jumlahkan seluruh ukuran dari objek atau benda kemudian dibagi dengan ukuran wadah atau *bin* yang ada. Namun hasil tersebut tidak dapat langsung digunakan karena jika diterapkan secara langsung jumlah yang dihasilkan *Lower Bound* tidak sesuai dengan hasil sebenarnya. Untuk itu diperlukan teknik atau metode yang mampu menyelesaikan permasalahan tersebut sehingga mampu memberikan hasil yang optimal dan sesuai dengan hasil sebenarnya.

Metaheuristic merupakan metode dalam kecerdasan buatan (*artificial intelligence*) yang mampu menyelesaikan permasalah-permasalah optimisasi secara efisien dan cepat [2]. Metode-metode metaheuristic yang dapat digunakan dalam menyelesaikan *Bin Packing Problem* adalah *Genetic Algorithm*, *Simulated Annealing*, *Tabu Search*, *Harmony Search*, *Particle Swarm Optimization*, *Firefly Algorithm*, dan lain sebagainya. Dalam tulisan ini, penulis akan menggunakan *Genetic Algorithm* dan *Firefly Algorithm* untuk menyelesaikan permasalahan *Bin Packing Problem*.

Genetic Algorithm atau Algoritma Genetika yang dikembangkan oleh John Holland pada tahun 1975, merupakan teknik penyelesaian permasalahan optimisasi yang mengambil terminologi dari teori evolusi dan genetika. Dimana didalamnya terdapat parameter-parameter bawaan dari genetika di teori evolusi antara lain seleksi alam, persilangan atau *crossover* dan mutasi gen.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Firefly Algorithm atau Algoritma Kunang-Kunang yang dikembangkan oleh Xin-She Yang sekitar tahun 2009 di *Cambridge University*. *Firefly Algorithm* merupakan metode yang termasuk kedalam *Swarm Intelligent* yang mengambil perilaku dari sekelompok objek makhluk hidup seperti hewan, dimana dalam metode ini Xin-She Yang terinspirasi dari perilaku sekumpulan kunang-kunang yang didasarkan pada pola kedipan cahaya yang dihasilkan oleh kunang-kunang tersebut [3].

Berdasarkan penjelasan diatas penulis akan membahas komparasi atau perbandingan hasil yang diberikan dari metode *Genetic Algorithm* dan *Firefly Algorithm* dalam menyelesaikan permasalahan *Bin Packing Problem*. *Bin Packing Problem* yang digunakan adalah satu dimensi dimana ukuran objek dan benda direpresentasikan dengan satu nilai integer (bilangan bulat) saja.

II. TINJAUAN PUSTAKA

A. *Bin Packing Problem*

Bin Packing Problem merupakan suatu permasalahan *NP-Complete* yang diambil atau dikembangkan dari *Knapsack Problem*. Dimana permasalahan ini menggambarkan sekumpulan objek S_i dengan ukuran yang berbeda-beda yang akan dimasukkan atau dipaket kedalam wadah atau bin L_i yang ukurannya telah ditentukan. Sebagai contoh diberikan sejumlah objek dengan ukuran S_1, S_2, S_3, …, S_n dan diberikan *bin* L_i dengan ukuran m. Setiap benda harus ditempatkan kedalam *bin* sehingga besar total semua benda tidak melebihi setiap *bin* dan memberikan jumlah *bin* yang minimum. Persamaannya adalah sebagai berikut [1].

$$\textbf{Subject to} \sum_{j=1}^{n} S_j x_{ij} \leq L_i y_i, \quad i \in N = \{1, \dots, n\},$$

$$\sum_{i=1}^{n} y_i, \qquad \sum_{j=1}^{n} x_{ij} = 1, \quad j \in N.$$

B. *Lower Bound*

Lower Bound merupakan batas indeks nilai terkecil yang digunakan untuk menghitung kompleksitas pada *Bin Packing Problem*. Untuk model *Bin Packing Problem* persamaan yang digunakan untuk mengitung *Lower Bound* *L* adalah sebagai berikut [1].

$$L = \left\lceil \frac{\sum_{j=1}^{n} S_j}{m} \right\rceil$$

C. *Genetic Algorithm*

Genetic Algorithm atau Algoritma Genetika yang diperkenalkan oleh John Holland seorang pendiri *Computing Evolution* di tahun 1975 merupakan suatu metode atau algoritma yang diambil dari evolusi alami dan genetika. Dimana didalamnya terdapat urutan langkah-langkah prosedur kromosom buatan yang bergerak dari satu populasi ke populasi baru [4]. Genetic Algorithm digunakan untuk memecah suatu pencarian nilai dalam sebuah masalah optimasi yaitu permasalahan yang tak linier [5].

Genetic Algorithm memakai mekanisme seleksi alam dan ilmu genetika sehingga istilah-istilah pada *Genetic Algorithm* akan bersesuaian. Sebuah solusi yang dibangkitkan dalam *genetic algorithm* sebagai kromosom, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, symbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan [5].

Berikut ini adalah tahapan-tahapan proses dalam Genetic Algorithm [4][6][7]:

1. Input Data, data yang didefinisikan sebelumnya dimasukkan kedalam proses.
2. Inisialisasi Populasi, data yang telah didefinisikan dan dimasukkan kedalam proses di inisialisasi dan dibangkitkan sebagai populasi awal dalam bentuk sejumlah individu atau kromosom secara acak.
3. Menghitung Nilai Fitness, setelah populasi dibentuk dari sekumpulan individu atau kromosom secara acak, dihitung nilai fitness masing-masing individu.
4. Seleksi, merupakan proses untuk memilih sejumlah individu yang memiliki nilai fitness terbaik dengan metode-metode seleksi yang ada untuk dijadikan induk atau parent untuk proses selanjutnya.
5. Crossover atau Persilangan, merupakan proses untuk memilih posisi string dari hasil seleksi dan menukarnya secara acak pada induk-induk yang akan menghasilkan offspring atau anak yang menjadi individu baru.
6. Mutasi, merupakan proses untuk mengembalikan materi genetic yang hilang dengan cara mengganti, menggeser, atau membalik nilai baik biner, string, numerik atau yang lainnya yang terpilih secara acak.

Parameter-parameter genetika yang digunakan dalam Genetic Algorithm berperan mengendalikan operator-operator genetika yang digunakan dalam proses optimasi. Berikut ini adalah parameter-parameter yang digunakan:

1. Jumlah Generasi (*G*), merupakan jumlah atau banyaknya proses iterasi yang digunakan genetic algorithm.
2. Ukuran Populasi (*N*), merupakan jumlah atau banyaknya individu atau kromosom pada satu populasi.
3. Probabilitas Crossover (*Pc*), merupakan persentase banyaknya kromosom yang akan dipindah silangkan.
4. Probabilitas Mutasi (*Pm*), merupakan persentase jumlah gen pada populasi yang akan dimutasi.

D. *Firefly Algorithm*

Firefly Algorithm merupakan salah satu dari algoritma metaheuristic terbaru untuk permasalahan optimisasi. Algoritma ini dikembangkan oleh Xin-She Yang pada tahun 2009 di Cambridge University. Algoritma ini terinspirasi oleh perilaku kedip cahaya pada kunang-kunang dimalam hari. Adapun tahapan proses algoritmanya adalah pembangkitan solusi secara acak akan dianggap sebagai kunang-kunang, dan kecerahan cahaya yang ditentukan merupakan performansi fungsi objektifnya [3][8].

Ada tiga aturan dalam algoritma ini, pertama semua kunang-kunang bersifat unisex, yang artinya beberapa kunang-kunang bisa tertarik pada satu atau lebih cahaya. Kedua, tingkat kecerahan pada kunang-kunang ditentukan dari pengkodean fungsi objektif. Ketiga adalah daya tarik berbanding lurus dengan kecerahan tetapi berbanding terbalik dengan jarak, dan kunang-kunang akan bergerak menuju yang paling terang, jika tidak ada yang paling terang maka kunang-kunang akan bergerak secara acak [8].

Pada fisika dasar menjelaskan bahwa intensitas cahaya berbanding terbalik dengan kuadrat jarak (r) dari sumber. Cahaya melewati media dengan koefisien penyerapan cahaya dari λ intensitas cayaha, I bervariasi dengan jarak r. Berikut persamaannya:

$$I(r) = I_0 e^{-\lambda r}$$

Dimana I_0 adalah intensitas dari titik sumber. Persamaan ini bisa dikombinasikan sebagai berikut:

$$I(r) = I_0 e^{-\lambda r^2}$$

Dikarenakan menggunakan $1/(1+\lambda r^2)$ lebih mudah mengitungnya dibandingkan menggunakan $e^{-\lambda r}$. Maka intensitas dapat dihitung menggunakan:

$$I(r) = \frac{I_0}{1 + \lambda r^2}$$

Daya tarik kunang-kunang dapat didefinisikan dengan persamaan sebagai berikut:

$$A(r) = \frac{A_0}{1 + \lambda r^2}$$

Berikut ini adalah urutan proses dari *Firefly Algorithm* [8]:

1. Bangkitkan Solution Set (kumpulan solusi) secara acak, $\{x_1, x_2, \ldots, x_k\}$
2. Hitung intensitas untuk masing-masing anggota solusi, $\{I_1, I_2, \ldots, I_k\}$
3. Pindahkan masing-masing kunang-kunang i menuju kunang-kunang yang memiliki cahaya paling cerah, dan jika tidak ada kunang-kunang yang paling cerah, pindahkan kunang-kunang tersebut secara acak.
4. Perbaharui Solution Set.
5. Hentikan proses jika kriteria penghentian terpenuhi, jika tidak kembali ke langkah 2.

Berikut ini adalah parameter-parameter pada *Firefly Algorithm* [3]:

1. α_t sebagai pengontrol keacakan (Randomness Control).
2. α_0 sebagai inisialisasi factor sekala keacakan.
3. t sebagai penghitung iterasi
4. δ sebagai factor pendingin (cooling factor) nilainya diantara 0.95 sampai 0.98.
   berikut persamaannya:
   $$\alpha_t = \alpha_0 \delta^t, \qquad 0 < \delta < 1$$
5. β sebagai pengendali daya tarik (*Control Attractiveness*).
6. L sebagai rata-rata skala masalah yang akan diselesaikan.
7. γ sebagai koefisien penyerapan cahaya.
8. n sebagai ukuran populasi kunang-kunang (jumlah kawanan) / *Swarm Size*.

III. PEMBAHASAN

Pada penelitian ini, data yang digunakan adalah 180 data *integer*/bilangan bulat yang merepresentasikan nilai atau ukuran dari objek. Data ini diambil dari data *benchmark* pada jurnal Burke, Hyde dan Kendell [9]. Berikut data yang digunakan.

```
56 53 56 49 41 51 55 51 52 48 47 45 47 53 49 48 43 46 48
48 48 53 53 48 48 46 52 53 43 48 51 53 46 44 50 53 53 57
52 51 56 50 42 44 58 48 59 43 49 53 43 50 59 49 53 47 52
47 53 44 58 50 41 48 53 48 52 46 45 52 52 51 46 54 55 54
56 54 56 41 37 54 51 46 49 55 43 50 57 57 43 49 50 55 41
49 46 47 54 50 46 55 59 47 47 50 53 47 57 50 49 54 53 44
48 51 55 53 54 49 47 64 49 60 56 47 48 48 50 38 45 43 47
58 45 45 72 55 49 50 49 50 51 55 53 47 49 54 53 59 51 56
56 43 53 52 53 52 53 45 57 49 47 45 47 60 44 53 51 50 50
54 65 47 50 49 46 45 47 57
```

Dan pada penelitian ini juga akan dilakukan 3 kali pengujian. Setiap pengujian ukuran *bin* yang digunakan berbeda yaitu 100, 110, dan 120.

Untuk metode Lower Bound tetap menggunakan persamaan

$$\boldsymbol{L} = \left\lceil \frac{\sum_{j=1}^{n} \boldsymbol{S_j}}{\boldsymbol{m}} \right\rceil$$

dimana didapat hasil sebagai berikut,

1. Untuk pengujian pertama didapat hasil:
   $$L = \left\lceil \frac{9066}{100} \right\rceil$$
   $$L = \lceil 90.66 \rceil$$
   $$L = 91$$
   Dengan demikian *bin* yang dihasilkan adalah 91 *bin*.
2. Untuk pengujian kedua didapat hasil:
   $$L = \left\lceil \frac{9066}{110} \right\rceil$$
   $$L = \lceil 82.41818 \rceil$$
   $$L = 83$$
   Dengan demikian *bin* yang dihasilkan adalah 83 *bin*.
3. Untuk pengujian ketiga didapat hasil:

e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$L = \left\lceil \frac{9066}{120} \right\rceil$$
$$L = \lceil 75.55 \rceil$$
$$L = 76$$

Dengan demikian *bin* yang dihasilkan adalah 76 *bin*.

Untuk pengujian yang dilakukan pada Genetic Algorithm akan dilakukan 10 percobaan dengan parameter-parameter sebagai beriktu:

1. Jumlah Generasi (*G*) yang digunakan adalah 1000.
2. Ukuran Populasi/Kromosom (*N*) yang digunakan adalah 100.
3. Probabilitas *Crossover* (*Pc*) yang digunakan adalah 0.8.
4. Probabilitas Mutasi (*Pm*) yang digunakan adalah 0.6.
5. Metode Seleksi yang digunakan adalah *Roulette Wheel Selection*.
6. Metode Crossover yang digunakan adalah *Permutation Crossover*.
7. Metode Mutasi yang digunakan adalah *Swapping Mutation*.

Untuk pengujian pertama dengan ukuran *bin* 100 disajikan dalam bentuk table dibawah ini.

Table 1 Hasil Pengujian Pertama untuk GA

Percobaan ke-	Hasil
1	92
2	93
3	93
4	92
5	92
6	93
7	94
8	92
9	92
10	93

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 92.6 *bin*.

Untuk pengujian kedua dengan ukuran *bin* 110 disajikan dalam bentuk table dibawah ini.

Table 2 Hasil Pengujian Kedua untuk GA

Percobaan ke-	Hasil
1	86
2	85
3	86
4	85
5	85
6	86
7	84
8	85
9	84
10	84

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 85 *bin*.

Untuk pengujian ketiga dengan ukuran *bin* 120 disajikan dalam bentuk table dibawah ini.

Table 3 Hasil Pengujian Ketiga untuk GA

Percobaan ke-	Hasil
1	79
2	79
3	78
4	77
5	78
6	78
7	77
8	79
9	79
10	78

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 78.2 *bin*.

Untuk pengujian yang dilakukan pada *Firefly Algorithm* akan dilakukan 10 percobaan dengan parameter sebagai berikut:

1. $\alpha_t = 0.2$
2. $t = 1000$
3. $\delta = 0.98$.
4. $\beta = 2$
5. $\gamma = 1$.
6. $n = 20$.

Untuk pengujian pertama dengan ukuran *bin* 100 disajikan dalam bentuk table dibawah ini.

Table 4 Hasil Pengujian Pertama untuk FA

Percobaan ke-	Hasil
1	93
2	92
3	92
4	93
5	93
6	92
7	92
8	92
9	92
10	92

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 92.3 *bin*.

Untuk pengujian kedua dengan ukuran *bin* 110 disajikan dalam bentuk table dibawah ini.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 5 Hasil Pengujian Kedua untuk FA

Percobaan ke-	Hasil
1	85
2	85
3	85
4	84
5	85
6	84
7	85
8	84
9	84
10	84

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 84.5 *bin*.

Untuk pengujian ketiga dengan ukuran *bin* 120 disajikan dalam bentuk table dibawah ini.

Table 6 Hasil Pengujian Ketiga untuk FA

Percobaan ke-	Hasil
1	79
2	79
3	78
4	77
5	77
6	77
7	77
8	77
9	78
10	77

Jika dirata-ratakan dari 10 percobaan diatas didapatlah hasil 77.6 *bin*.

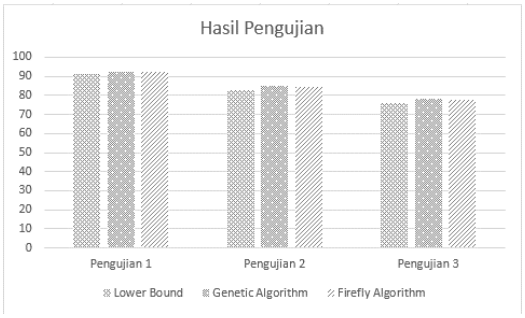
Dari seluruh pengujian yang dilakukan dapat dilihat hasil bahwa lower bound masih lebih baik dibandingkan GA dan FA. Namun *lower bound* merupakan penyelesaian permasalahan secara *polynomial*, tidak dapat langsung diterapkan. Sehingga yang akan dibahas adalah perbandingan hasil dari Genetic Algorithm dan Firefly Algorithm.

Pada pengujian pertama dengan jumlah *bin* 100 *Firefly Algorithm* menghasilkan nilai lebih baik dibandingkan dengan *Genetic Algorithm* dengan perbandingan nilai 92.3:92.6. Walau perbandingannya hanya terpaut 0.3 namun dapat terlihat *Firefly Algorithm* masih lebih unggul dibandingkan *Genetic Algorithm*.

Pada pengujian kedua dengan jumlah *bin* 110 *Firefly Algorithm* masih menghasilkan nilai lebih baik dibandingkan dengan *Genetic Algorithm* dengan perbandingan nilai 84.5:85. Perbandingan yang dihasilkan terpaut 0.5 namun dapat dilihat *Firefly Algorithm* masih lebih unggul dibandingkan dengan *Genetic Algorithm*.

Pada pengujian kedua dengan jumlah *bin* 120 *Firefly Algorithm* tetap masih menghasilkan nilai lebih baik dibandingkan dengan *Genetic Algorithm* dengan perbandingan nilai 77.6:78.2. Perbandingan yang dihasilkan terpaut 0.6 namun dapat dilihat *Firefly Algorithm* masih lebih unggul dibandingkan dengan *Genetic Algorithm*.

Berikut ini adalah grafik hasil dari *Lower Bound*, *Genetic Algorithm* dan *Firefly Algorithm*.



Gambar 1 Grafik Hasil Pengujian

IV. KESIMPULAN

Dari hasil analisis, pengujian serta pembahasan diatas dapat disimpulkan bahwa:

1. Dari hasil pengujian yang dilakukan terhadap permasalahan bin packing problem satu dimensi, metode *Firefly Algorithm* masih lebih baik dibandingkan dengan *Genetic Algorithm* dilihat dari hasil pengujian yang dilakukan oleh peneliti.
2. Jika dibandingkan dengan *Lower Bound*, kedua metode tersebut masih belum bisa melampaui hasil yang diberikan oleh metode *lower bound*. Namun hasil kedua metode tersebut sudah hampir mendekati nilai pada *lower bound*.

REFERENSI

[1] Martello, S., & Toth, P., 1990, *Knapsack Problem: Algorithms and Computer Implementations*, John Wiley & Sons Ltd: England.
[2] Purnama, H. D., 2014, *Cara Mudah Belajar Metode Optimisasi Metaheuristik Menggunakan Matlab*, Gava Media: Yogyakarta.
[3] Yang, X., & He, X., 2013, *Firefly Algorithm: Recent Advances and Application*, International Journal of Swarm Intelligence 1.1, pp: 36-50.
[4] Negnevitsky, M., 2005, *Artificial Intelligence: A Guide to Intelligent Systems, Second Edition*, Addison-Wesley: Harlow.
[5] Gen, M., & Cheng, R., 1997, *Genetic Algorithms & Engineering*, John Willey & Sons Ltd: USA.

[6] Perdana, A., 2014, *Analisis Performansi pada Penerapan Hukum Ketetapan Hardy-Weinberg dalam Algoritma Genetika*, Thesis, Universitas Sumatera Utara: Medan.
[7] Panggabean, T. N., 2016, *Analisis Reduksi NP-Hard Bin Packing Problem dalam Algoritma Genetika pada Hukum Ketetapan Hardy-Weinberg*, Thesis, Universitas Sumatera Utara: Medan.
[8] Tilahun, S. L., & Ong, H. C., 2012, *Modified Firefly Algorithm*, Hindawi Publication Corporation, Journal of Applied Mathematics Vol. 2012.
[9] Burke, E. K., Hyde, M. R., and Kendall, G., 2010, *Provideing a Memory Mechanism to Enchance the Evolutionary Design of Heuristics*, Proceedings of the IEEE Wordl Congress on Computational Intelligence, Barcelona, pp. 3884-3890.
[10] Baldi, M. M., Crainic, T. G., Perboli, G., Tadei, R., 2014, *Branch-and-Price and Beam Search Algorithms for the Variable Cost and Size Bin Packing Problem with Optional Items*, ANNALS OF OPERATION RESEARCH, Vol. 222, pp. 125-141, Springer.