

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengolahan data dan analisis yang telah dilakukan, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Metode HGA-AVNS terbukti dapat diterapkan untuk menyelesaikan permasalahan *Resource-Constrained Project Scheduling Problem* (RCPSP) pada dataset PSPLIB instansi J601_1, J901_1, dan J1201_9. Metode ini menggabungkan *Genetic Algorithm* sebagai mekanisme eksplorasi *global* dan *Adaptive Variable Neighborhood Search* sebagai mekanisme eksploitasi lokal. GA berperan dalam membentuk berbagai alternatif urutan aktivitas, sedangkan AVNS digunakan untuk memperbaiki solusi terbaik melalui struktur *neighborhood swap*, *insert*, dan *reverse*.
2. Pada pengujian 30 replikasi dengan seed berbeda, HGA-AVNS menghasilkan rata-rata makespan sebesar 77,00 pada J601_1, 81,67 pada J901_1, dan 126,93 pada J1201_9, dengan rata-rata deviasi terhadap BKS masing-masing sebesar 0,00%, 11,87%, dan 13,33%. Pada J601_1, seluruh replikasi HGA-AVNS mencapai BKS sebesar 77. Pada J901_1 dan J1201_9, HGA-AVNS belum mencapai BKS, tetapi menghasilkan rata-rata makespan yang lebih rendah daripada VNS serta lebih rendah daripada GA standar. Hasil tersebut menunjukkan bahwa penambahan AVNS memberikan perbaikan kualitas solusi yang lebih terlihat pada instansi dengan jumlah aktivitas yang lebih besar.
3. Hasil uji Friedman menunjukkan adanya perbedaan makespan yang signifikan antara GA standar, VNS, dan HGA-AVNS pada J601_1,

J901_1, dan J1201_9 karena seluruh nilai *p-value* lebih kecil dari 0,05. Hasil uji Wilcoxon menunjukkan bahwa HGA-AVNS tidak berbeda signifikan dengan GA standar pada J601_1 dan J901_1, dengan *p-value* masing-masing sebesar 0,186 dan 0,079. Namun, HGA-AVNS lebih baik secara signifikan daripada GA standar pada J1201_9 dengan *p-value* sebesar 0,031. Dibandingkan VNS, HGA-AVNS menghasilkan makespan yang lebih rendah secara signifikan pada ketiga instansi. Dengan demikian, keunggulan HGA-AVNS terhadap GA standar belum konsisten pada seluruh instansi replikasi, tetapi semakin terlihat pada instansi J120 yang lebih kompleks.

4. Pada pengujian konfigurasi kedua terhadap 30 instansi PSPLIB, HGA-AVNS menghasilkan rata-rata deviasi terhadap BKS sebesar 0,55% pada kelompok J60, 2,52% pada J90, dan 6,32% pada J120. HGA-AVNS mencapai BKS pada delapan dari sepuluh instansi J60 dan dua dari sepuluh instansi J90, sedangkan pada kelompok J120 belum terdapat hasil yang mencapai BKS. Meskipun deviasi meningkat seiring bertambahnya ukuran dataset, HGA-AVNS menghasilkan rata-rata deviasi paling rendah dibandingkan GA standar dan VNS pada ketiga kelompok dataset. Uji Wilcoxon terhadap 30 instansi juga menunjukkan bahwa HGA-AVNS menghasilkan makespan yang lebih rendah secara signifikan daripada GA standar dengan *p-value* 0,001 dan daripada VNS dengan *p-value* kurang dari 0,001.
5. Dari sisi waktu komputasi, VNS merupakan metode dengan runtime paling rendah, diikuti oleh GA standar, sedangkan HGA-AVNS membutuhkan runtime paling tinggi. Peningkatan waktu komputasi HGA-AVNS terjadi karena metode ini tidak hanya menjalankan proses evolusi GA, tetapi juga mengevaluasi kandidat solusi tambahan

melalui pencarian lokal AVNS. Oleh karena itu, peningkatan kualitas solusi yang dihasilkan HGA-AVNS diperoleh dengan konsekuensi bertambahnya waktu komputasi.

6. Secara keseluruhan, penelitian ini menunjukkan bahwa integrasi AVNS ke dalam GA mampu meningkatkan kemampuan eksploitasi lokal dan menghasilkan solusi yang secara umum lebih dekat dengan BKS dibandingkan GA standar dan VNS. Keunggulan HGA-AVNS terlihat dari pencapaian BKS pada delapan instansi J60 serta rata-rata deviasi yang paling rendah pada kelompok J60, J90, dan J120. Selain itu, pengujian melalui replikasi dan beberapa instansi menunjukkan bahwa metode dapat diterapkan pada karakteristik RCPSP yang berbeda. Meskipun membutuhkan waktu komputasi lebih tinggi, HGA-AVNS tetap memberikan kontribusi positif sebagai alternatif metode yang memprioritaskan kualitas solusi dan minimisasi *makespan*.

6.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya dapat melakukan pengujian pada lebih banyak instansi PSPLIB, tidak hanya 10 instansi pada dataset J60, J90, dan J120. Pengujian pada lebih banyak instansi diperlukan agar kinerja HGA-AVNS dapat dievaluasi secara lebih menyeluruh pada berbagai tingkat kompleksitas RCPSP.
2. Penelitian selanjutnya dapat melakukan *tuning* parameter secara lebih mendalam, terutama pada parameter AVNS seperti jumlah iterasi AVNS, interval pemanggilan AVNS, *stagnation trigger*, bobot

neighborhood, dan persentase *Random immigrant*. Penyesuaian parameter tersebut berpotensi meningkatkan kualitas solusi sekaligus mengendalikan waktu komputasi.

3. Mekanisme AVNS dapat dikembangkan dengan menambahkan atau memodifikasi struktur *neighborhood* selain *swap*, *insert*, dan *reverse*. Penggunaan *neighborhood* yang lebih sesuai dengan karakteristik RCPSP dapat membantu algoritma dalam mengeksplorasi solusi yang lebih beragam dan memperbaiki kualitas *makespan*.
4. Penelitian selanjutnya dapat membandingkan HGA-AVNS dengan metode metaheuristik lain, seperti *Simulated Annealing*, *Particle Swarm Optimization*, *Ant Colony Optimization*, atau algoritma hibrida lainnya. Perbandingan tersebut diperlukan untuk mengetahui posisi kinerja HGA-AVNS terhadap metode *optimasi* lain dalam menyelesaikan RCPSP.
5. Dari sisi implementasi program, penelitian selanjutnya dapat melakukan *optimasi* kode untuk mengurangi *runtime*, terutama pada proses evaluasi SSGS dan pencarian lokal AVNS. *Optimasi* dapat dilakukan dengan memperbaiki struktur data, mengurangi evaluasi kandidat yang tidak diperlukan, atau menggunakan komputasi paralel agar waktu komputasi menjadi lebih efisien.

DAFTAR PUSTAKA

- Bierwirth, C., Mattfeld, D. C., & Kopfer, H. (1997). *On Permutation Representations for Scheduling Problems*.
- Brucker, P., & Knust, S. (1999). *2 SOLVING LARGE-SIZED RESOURCE-CONSTRAINED PROJECT 2.1 INTRODUCTION SCHEDULING PROBLEMS*.
- Coelho, J., & Vanhoucke, M. (2020). Going to the core of hard resource-constrained project scheduling instances. *Computers and Operations Research*, 121. <https://doi.org/10.1016/j.cor.2020.104976>
- Creemers, S. (2019). *The preemptive stochastic resource-constrained project scheduling problem: an efficient globally optimal solution procedure*.
- Fang, Y., & Li, J. (2010). A Review of Tournament Selection in Genetic Programming. Dalam *LNCS* (Vol. 6382). Springer-Verlag.
- Gendreau, M., & Potvin, J.-Y. (2010). *International Series in Operations Research & Management Science*. <http://www.springer.com/series/6161>
- Gu, X., Huang, M., & Liang, X. (2019). An improved genetic algorithm with adaptive variable neighborhood search for FJSP. *Algorithms*, 12(11). <https://doi.org/10.3390/a12110243>
- Habibi, F., Barzinpour, F., & Sadjadi, S. J. (2018). Resource-constrained project scheduling problem: review of past and recent developments. *Journal of Project Management*, 55–88. <https://doi.org/10.5267/j.jpm.2018.1.005>
- Harahap, R. F., & Muliono, R. (2025). The Application Of Genetic Algorithm In Construction Project Planning System At Cv. Haza Mulia Engineering. Dalam *JITE* (Vol. 8). <https://doi.org/10.31289/jite.v8i3Spc.14432>
- Hartmann, S. (1998). A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*, 45(7), 733–750. [https://doi.org/10.1002/\(SICI\)1520-6750\(199810\)45:7<733::AID-NAV5>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C)

- Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2018). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4), 2191–2233. <https://doi.org/10.1007/s10462-017-9605-z>
- Hussain, W., Trivedi, M. K., Kansal, R., & Student, P. G. (2015). Optimisation of Construction Resource Allocation and Levelling Using Genetic Algorithm. *International Journal of Innovative Research in Science, Engineering and Technology (An ISO, 3297)*. <https://doi.org/10.15680/IJIRSET.2015.0406073>
- Joshi, K., Jain, K., & Bilolikar, V. (2016). A VNS-GA-based hybrid metaheuristics for resource constrained project scheduling problem. Dalam *Int. J. Operational Research* (Vol. 27, Nomor 3).
- Klimek, M. (2010). A genetic algorithm for the project scheduling with the resource constraints. *Annales UMCS, Informatica*, 10(1). <https://doi.org/10.2478/v10065-010-0042-8>
- Kolisch, R. (1996). EUROPEAN JOURNAL OF OPERATIONAL RESEARCH Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. Dalam *European Journal of Operational Research* (Vol. 90).
- Kolisch, R., & Sprecher, A. (1996). PSPLIB-A project scheduling problem library. Dalam *European Journal of Operational Research* (Vol. 96).
- Liu, J., Liu, Y., Shi, Y., & Li, J. (2020). Solving the resource-constrained project scheduling problem by genetic algorithm. *Journal of Japan Industrial Management Association*, 57(6), 520–529. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000874](https://doi.org/10.1061/(asce)cp.1943-5487.0000874)
- Mladenović, N., & Hansen, P. (1997). VARIABLE NEIGHBORHOOD SEARCH. Dalam *Computer.* Ops Res* (Vol. 24, Nomor 1).
- Möhring, R. H., Schulz, A. S., Stork, F., & Uetz, M. (2003). *Solving Project Scheduling Problems by Minimum Cut Computations*.
- Naaman, D. W., Ahmed, B. T., & Ibrahim, M. I. (2025). Optimization by Nature: A Review of Genetic Algorithm Techniques. *The Indonesian Journal of Computer Science*, 14(1). <https://doi.org/10.33022/ijcs.v14i1.4596>

- Nonobe, K., & Ibaraki, T. (t.t.). *Tabu Search Algorithms for Extended Models of the Resource-Constrained Project Scheduling Problem*.
- Nonobe, K., & Ibaraki, T. (2002). *25 FORMULATION AND TABU SEARCH ALGORITHM FOR THE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM*.
- Pardo, A. G., & Camacho, D. (2014). *A New CSP Graph-Based Representation to Resource-Constrained Project Scheduling Problem*. IEEE.
- Pellerin, R., Perrier, N., & Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. Dalam *European Journal of Operational Research* (Vol. 280, Nomor 2, hlm. 395–416). Elsevier B.V. <https://doi.org/10.1016/j.ejor.2019.01.063>
- Pimapunsi, K., Weeranant, D., & Riel, A. (2020). *GENETIC ALGORITHMS FOR THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM IN AIRCRAFT HEAVY MAINTENANCE*.
- Popescu, D. A. (2025). An Enhanced Genetic Algorithm for Optimized Educational Assessment Test Generation Through Population Variation. *Big Data and Cognitive Computing*, 9(4). <https://doi.org/10.3390/bdce9040098>
- Roy, B., & Sen, A. K. (2019). Meta-heuristic techniques to solve resource-constrained project scheduling problem. Dalam *Lecture Notes in Networks and Systems* (Vol. 56, hlm. 93–99). Springer. https://doi.org/10.1007/978-981-13-2354-6_11
- Valls, V., Ballestín, F., & Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 185(2), 495–508. <https://doi.org/10.1016/j.ejor.2006.12.033>
- Valls, V., Quintanilla, S., & Ballestín, F. (2003). Resource-constrained project scheduling: A critical activity reordering heuristic. *European Journal of Operational Research*, 149(2), 282–301. [https://doi.org/10.1016/S0377-2217\(02\)00768-3](https://doi.org/10.1016/S0377-2217(02)00768-3)

- Viana, M. S., Junior, O. M., & Contreras, A. R. C. (2020). An improved local search genetic algorithm with a new mapped adaptive operator applied to Pseudo-coloring problem. *Symmetry*, 12(10), 1–36. <https://doi.org/10.3390/sym12101684>
- Zamani, R. (2013). *Integrating Iterative Crossover Capability in Orthogonal Neighborhoods for Scheduling Resource-Constrained Projects*.