

Enhancing Operational Efficiency: Harnessing Metaheuristic Algorithms for Scheduling in Cellular Manufacturing

Jayasuriya E¹, Sanjay Krishna R², Prakash R³ and Hemashree P⁴

¹⁻⁴Coimbatore Institute of Technology, Coimbatore, India

Email: {jayasuriya352002, sanjaykrishna2001, prakashr7d}@gmail.com, hemashree@cit.edu.in

Abstract. In Cellular Manufacturing (CM) scheduling, a set of components must be processed in cells by a set of machines. One of the key decisions in cell production is timing of components in each cell. The goal is to find the sequence of performing pieces in each group and the sequence of performing groups of pieces in cells on a set of machines; in such a way that the desired criterion is optimized in the schedule. This proposed work explores various meta-heuristic approaches to solve the CM scheduling problem and compare it with the existing state-of-the-art Genetic Algorithm. The linear problem of scheduling for 6 jobs along with their completion time in 3 cells using meta-heuristic algorithms is implemented. The algorithms are evaluated based on fitness score computed in terms of completion time and load balancing score. In this minimization problem, the different proposed algorithms have outperformed the Genetic algorithm in various aspects. Particle swarm optimization has found to have the minimum fitness score.

Keywords—Cellular Manufacturing, Scheduling Optimization, Simulated Annealing, Genetic Algorithm, Particle Swarm Optimization, Adaptive Mutation, Tabu Search, Duelist Algorithm

I. INTRODUCTION

Cellular manufacturing is a manufacturing approach that organizes production into cells, each of which is dedicated to producing a specific family of products. The goal of cellular manufacturing is to improve efficiency and reduce lead times by minimizing the distance that materials and products have to travel between workstations. Schedule optimization is an important problem in cellular manufacturing. The goal is to assign tasks to workstations in a way that minimizes the total processing time, while ensuring that the workload is balanced and that the capacity of each workstation is not exceeded.

This can lead to a variety of benefits, such as, Increased Productivity: By optimizing schedules, companies can reduce idle time and improve resource utilization, which can lead to increased productivity and output. Improved Customer Satisfaction: Optimized schedules can help ensure that products or services are delivered on time and with consistent quality, which can lead to improved customer satisfaction and loyalty. Cost Savings: By optimizing schedules, companies can reduce the need for overtime, eliminate bottlenecks, and minimize waste, which can lead to significant cost savings. Better Decision-Making: Schedule optimization can provide decision-makers with insights into how changes in resources or production processes can impact overall performance, enabling them to make more informed decisions. Improved Safety: Optimized schedules can help reduce the risk of accidents and injuries by ensuring that workers are not overworked or fatigued.

This article delves into the realm of cellular manufacturing optimization and explores the application of various metaheuristic algorithms, including Simulated Annealing, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Adaptive Mutation Genetic Algorithm (AMGA), Tabu Search (TS), and the Duelist Algorithm. By harnessing the strengths of these algorithms, manufacturers can significantly enhance their decision-making processes and operational efficiency.

II. RELATED WORK

In [1], the authors have proposed an improved algorithm for layout design in cellular manufacturing systems. The authors develop a modified version of the cell formation algorithm that takes into account the effects of the layout on the manufacturing system's performance. The proposed algorithm uses an iterative approach that optimizes the layout by minimizing the sum of the weighted distances between cells. The results of the simulation showed that the proposed algorithm outperformed the traditional cell formation algorithm in terms of layout performance. Analogously in [2], the authors have proposed a Genetic Algorithm approach to solve a cell formation problem in cellular manufacturing. The study aimed to identify the most effective cell configuration that minimizes intercellular movement, setup time, and workload imbalance. The authors tested the algorithm on a real-world manufacturing dataset and compared it to other cell formation techniques. The results showed that the genetic algorithm was superior to the other methods in terms of the quality of the cell formation. Similarly, in [3], the authors have proposed the Bacteria Foraging Optimization (BFO) algorithm to solve a cell formation problem in cellular manufacturing systems. The proposed algorithm aims to minimize the total intercellular movement and workload imbalance. The authors compared their algorithm to other optimization techniques, and the results showed that the BFO algorithm outperformed the other methods in terms of solution quality and computational time. In [4], the authors have proposed a GA and Memetic Algorithm (MA) to sequence and schedule cellular manufacturing systems. The study aims to minimize the total tardiness, the total number of setups, and the maximum workload among the cells. The authors compared the performance of the two algorithms to each other and to other scheduling techniques. The results showed that the MA outperformed the GA and the other scheduling techniques in terms of the quality of the solutions. Along the same lines, in [5], the authors have proposed an Ant Colony Optimization (ACO) algorithm to solve the inter-cell layout problem in cellular manufacturing systems. The study aims to minimize the total distance traveled by parts within and between cells. The authors compared the algorithm to other optimization techniques and showed that the ACO algorithm outperformed the other methods in terms of solution quality and computational time. In [6], a new CMS model that incorporates type-2 fuzzy variables to improve the accuracy of the optimization process by minimizing five different costs associated with CMS while following the specified constraints was proposed. They evaluated their model using three meta-heuristic algorithms: the Particle Swarm Optimization (PSO), the Firefly Algorithm (FA), and the Differential Evolution Algorithm (DE) after fine tuning their parameters using Taguchi method. The authors show that their model outperforms existing models and that the PSO algorithm produces the best results. In [7], a new mathematical model (TPM), to minimize cycle time, is compared with existing model TGM, and proposes Meta-Heuristic based algorithms namely the Local Search (LS), Stochastic Local Search (SLS) and Adaptive Local Search (ALS) algorithms on small and large sized problems to solve large-size flexible robotic cell problems. The authors compare their algorithm with existing algorithms, and the results show that the ALS algorithm produces the best results in terms of solution quality and computation time. In [8], the authors have analyzed various algorithms, including GA, ACO, and TS, and evaluated their effectiveness in solving Cellular Manufacturing System (CMS) problems. The authors conclude that meta-heuristic algorithms are effective in solving CMS problems and that GA produce the best results. In [9], the authors have proposed a novel two-stage heuristic algorithm to optimize in-plant material supply in CMS. They compare it with CPLEX optimizer which is the state-of-the-art commercial solver. The authors show that their algorithm can improve the sustainability and mass customization capabilities of CMS. In [10], the authors have focused on using GA to solve scheduling problems in CMS. The objective was to minimize the total time of completion of product and transitions between cells to determine the sequence making this a multi-criterion mixed integer programming model. The authors show that their model can optimize the scheduling of CMS and reduce production costs.

III. MOTIVATION

In today's highly competitive manufacturing landscape, the pursuit of productivity, cost-effectiveness, decision making, and safety has become paramount. To meet these demands, manufacturers are constantly seeking

innovative approaches to optimize their operations. One such approach gaining prominence is the utilization of metaheuristic optimization algorithms in cellular manufacturing systems. These algorithms offer a powerful means of achieving optimal solutions by efficiently exploring large solution spaces and finding near-optimal solutions within reasonable timeframes. By exploring the capabilities and applications of these metaheuristic optimization algorithms, manufacturers can unlock new dimensions of efficiency, cost reduction, and productivity in cellular manufacturing systems. The subsequent sections of this article will delve into the methodologies that highlight the effectiveness and benefits of these algorithms in various manufacturing contexts.

IV. PROBLEM DOMAIN

Cellular manufacturing is a production system that organizes manufacturing processes into small, self-contained units called cells. Each cell is responsible for producing a specific group of products or components. The concept of cellular manufacturing is derived from the principles of lean manufacturing and aims to improve efficiency, productivity, and quality in the manufacturing process. In a Cellular Manufacturing System (CMS), the layout of the production floor is designed to group together machines, equipment, and workers that are required to complete a particular set of tasks. This arrangement eliminates the need for excessive material handling and transportation, as well as reduces the distance between operations, resulting in smoother and more streamlined production flows.

V. PROBLEM DEFINITION

The field of Cellular Manufacturing has complex design and is hard to solve for NP-Hard Cellular manufacturing systems. Some of the CMS problems include minimizing the cycle time, optimizing material supply, and scheduling problems. Cellular manufacturing often involves job shop scheduling, where a variety of products with different routings and processing requirements are produced. Determining the sequence of operations, allocating resources, and optimizing the schedule to minimize make span (total time to complete all jobs) and maximize throughput can be challenging due to the complexity and variability of job requirements. Achieving a balanced workload across cells is crucial to avoid bottlenecks or underutilization of resources. Scheduling tasks and jobs in a way that evenly distributes work among cells can be complex, especially when dealing with varying task durations, priorities, and capacities of cells.

VI. SOLUTION METHODOLOGIES:

The main focus is to explore different possible meta-heuristic approaches to the problem of schedule optimization in Cellular Manufacturing Systems (CMS) like Swarm Based Approach, Mixed models' approach and Reinforcement learning based approach as there are only a very few studies available. Genetic algorithm is used as a state-of-the-art benchmark model to compare the performance of the Metaheuristic algorithms with. The following meta-heuristic methodologies are proposed to establish our findings.

Simulated Annealing (SA) [11] is a probabilistic optimization algorithm that is used to find the global minimum of a complex cost or objective function. It is often used in optimization problems that involve a large search space or a high number of local minima. The algorithm is inspired by the annealing process used in metallurgy, where a metal is heated and slowly cooled to reduce its defects and increase its strength. In SA, the algorithm starts with an initial solution and then iteratively generates a new solution by randomly perturbing the current solution. The perturbation is controlled by a temperature parameter, which is gradually decreased over time to allow the algorithm to converge to a better solution. The probability of accepting a worse solution at a higher temperature is given by the Metropolis-Hastings criterion, which balances the exploration of new solutions with the exploitation of the best solution found so far.

SA is a powerful optimization technique that has been successfully applied in many fields, including machine learning, computational physics, and engineering. It can be used to solve a wide range of optimization problems, including clustering, scheduling, and routing. However, it can be computationally expensive, and the quality of the solution depends on the choice of the initial temperature and the cooling schedule.

Simulated Annealing (SA) and Genetic Algorithm (GA) are two different optimization algorithms that can be used separately to solve complex optimization problems. However, in some cases, combining these two algorithms can result in a more effective optimization approach. The idea of combining SA and GA is to use the strengths of both algorithms to improve the overall optimization process. SA is good at escaping local optima and exploring the search space, while GA is good at exploiting promising regions of the search space. The basic

idea of combining these two algorithms is to use SA as a local search technique within each generation of GA. In this approach, the GA is used to generate a set of candidate solutions, and then SA is used to optimize each candidate solution locally. The combination of SA and GA can be done in different ways, depending on the problem being solved. One common approach is to use SA to refine the offspring generated by GA. Another approach is to use SA to optimize the population of solutions in GA after a certain number of generations. The combination of SA and GA can be effective in solving complex optimization problems, especially when the problem has a large search space and many local optima. However, this approach can be computationally expensive, and the quality of the solution depends on the choice of the parameters and the optimization approach used.

Particle Swarm Optimization (PSO) [12] is a population-based stochastic optimization algorithm that is used to find the optimal solution to a given optimization problem. It is inspired by the collective behavior of social animals such as birds and fish, where individuals interact with each other to achieve a common goal. In PSO, a population of particles is initialized with random positions and velocities in the search space. Each particle represents a potential solution to the problem being solved. The particles move through the search space, guided by their own best position and the best position found by the swarm so far. The best position found by each particle is updated as the particle moves through the search space. The position and velocity of each particle are updated in each iteration of the algorithm based on the best position found by the particle and the best position found by the swarm. The update is performed according to a set of rules that determine how the particles move through the search space. The update rule is based on the current position, velocity, and the best position found so far. The PSO algorithm is effective in solving optimization problems that involve a large search space and multiple local optima. It is often used in engineering, computer science, and other fields to solve a wide range of problems, including clustering, scheduling, and optimization of complex functions. The main advantage of PSO over other optimization algorithms is its ability to explore the search space efficiently and escape from local optima. However, the quality of the solution obtained by PSO depends on the choice of the algorithm parameters, such as the number of particles, the maximum velocity, and the acceleration constants.

Adaptive Mutation Genetic Algorithm (AMGA) [13] is a variation of the standard GA that incorporates an adaptive mutation operator. The mutation operator is responsible for introducing new genetic information into the population, which can help the algorithm explore new areas of the search space and escape from local optima. In standard GA, the mutation operator is typically set to a fixed value throughout the optimization process. However, in AMGA, the mutation operator is dynamically adjusted based on the performance of the algorithm. The goal of the adaptive mutation is to balance exploration and exploitation in the search space and improve the convergence rate of the algorithm. The adaptive mutation operator in AMGA is typically controlled by a parameter that is updated based on the fitness of the individuals in the population. If the algorithm is converging slowly, the mutation rate is increased to encourage exploration of new regions of the search space. On the other hand, if the algorithm is converging quickly, the mutation rate is decreased to encourage exploitation of promising regions. The AMGA algorithm has been shown to be effective in solving complex optimization problems, especially those with multiple local optima. The adaptive mutation operator allows the algorithm to efficiently explore the search space while maintaining a good balance between exploration and exploitation. However, the performance of AMGA depends on the choice of the mutation operator and the adaptive parameter update strategy. Careful tuning of these parameters is necessary to ensure optimal performance of the algorithm.

Simulated Annealing (SA) and Tabu Search (TS) are two popular metaheuristic optimization algorithms used to solve combinatorial optimization problems. These two methods are often combined to improve their effectiveness and efficiency. SA is a stochastic optimization algorithm that is inspired by the physical process of annealing. The algorithm begins with an initial solution and then explores the solution space by iteratively modifying the solution. The algorithm accepts modifications that improve the solution, but also accepts modifications that worsen the solution with a certain probability. This probabilistic acceptance allows the algorithm to avoid getting trapped in local optima and explore the solution space more extensively. TS, on the other hand, is a metaheuristic optimization algorithm that uses a memory-based approach to avoid revisiting previously explored solutions. This memory is known as a tabu list, and it prevents the algorithm from revisiting solutions that have already been visited. The tabu list also contains information about the previous moves made by the algorithm, which helps to guide the search towards promising areas of the solution space. When these two algorithms are combined, the resulting SA with TS algorithm takes advantage of the strengths of both methods. The SA component allows the algorithm to explore the solution space more effectively, while the TS component prevents revisiting previously explored solutions and guides the search towards promising areas of the solution space. In summary, SA with TS is a powerful optimization algorithm that can be used to solve a wide range of

problems. Its effectiveness and efficiency make it a popular choice for solving complex combinatorial optimization problems.

The Duelist Algorithm (DA) [14] is a Reinforcement Learning algorithm that was introduced by OpenAI in 2019. It is a form of two-player game where both players use Deep Neural Networks to play against each other. The goal of the algorithm is to learn a policy for both players, such that each player tries to maximize their own reward. DA is based on the popular Deep Reinforcement Learning algorithm, Deep Q-Network (DQN). However, unlike DQN, the DA separates the value and advantage functions, and uses them to calculate the Q-values of each action. This separation allows the algorithm to learn more efficiently and improve the stability of the training process. The Duelist algorithm uses a form of prioritized experience replay, where it prioritizes the experiences that have the most impact on the Q-values. This helps the algorithm to learn from the most important experiences, and improves its ability to generalize to new situations. Overall, DA has shown promising results in various environments, including playing Atari games and continuous control tasks. It is a powerful technique for learning complex behaviors and can be applied to a wide range of problems.

VII. EXPERIMENTAL SETUP

The experimentation was done using python, initially parameters such as the number of operations, number of cells, population size, number of generations are defined. The algorithms are implemented for 6 jobs which is the number of operations. The processing time for each of the six jobs is introduced. A total of 3 cells are initialized. Jobs have to be assigned in such a way that it minimizes the fitness score. A population of size 50 is generated each time. The number of generations is set to 100 which is the termination condition too. A fitness function was defined to evaluate the fitness of a solution based on total completion time and load balancing. It calculated the completion times for each operation in each cell and determined the total completion time and load balancing score. Next, we initialize a random population and pass it to the heuristic algorithm to find the best solution. AMGA works by GA functions include generating a random chromosome representing the scheduling of operations in cells, performing single-point crossover between parent chromosomes, and applying mutation to chromosomes based on a mutation rate. The mutation rate was dynamically determined using the fitness and average fitness of the population. GA was then executed for the specified number of generations. In each generation, fitness was calculated for each chromosome, parents were selected for crossover based on their fitness, crossover and mutation were performed to create new child chromosomes, and the old population was replaced with the new population. The mutation rate was updated by decaying it using the mutation rate decay factor. Similarly, factors for other meta heuristic algorithms such as temperature, cooling rate for SA, inertia, cognitive factor and social factor for PSO, mutation rates for AMGA, tabu list and tabu tenure for TS algorithm and number of duels for DA were initialized. The steps appropriate for each of the algorithms were implemented. After the completion of the algorithm, the best solution was selected from the final population based on its fitness. Finally, the best operation scheduling in cells (the best solution) and the corresponding total completion time is found.

TABLE I COMPARISON OF RESULTS

Algorithm	Fitness Score
Simulated Annealing	72
Simulated Annealing + Genetic Algorithm	34
Particle Swarm optimization	29
Adaptive Mutation Genetic Algorithm	46
Simulated Annealing + Tabu Search	35
Duelist Algorithm	112

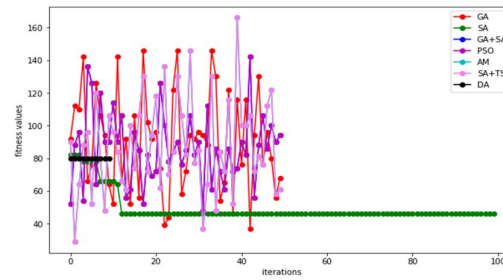


Figure 1. Comparison of Algorithms

VIII. RESULTS AND SENSITIVITY ANALYSIS:

Calculation of fitness function is done based on two criteria namely the total completion time and the load balancing. Total completion time is the time taken to complete all the operations in the manufacturing process. The function calculates this by iterating through the list of operations, calculating the time it takes to complete each operation in the cell assigned to it, and then finding the maximum time among all the cells. Load balancing is a measure of how evenly the workload is distributed among the cells. The function calculates this by finding the difference between the maximum processing time and the minimum processing time among all the cells. At the end of 100 iterations for each algorithm the solution in the population with minimum fitness score is assigned to be the best solution. The algorithms and their best fitness score recorded are presented in Table 1.

IX. COMPARISON OF RESULTS:

A Plot on the fitness score of the algorithms with respect to the number of iterations is generated to enable ease of comparison as in Fig. 1. The results obtained from the experiments demonstrate the effectiveness and efficiency of these algorithms in improving the overall performance of the manufacturing system. Among the algorithms considered, SA achieved a fitness score of 72, indicating its ability to provide satisfactory solutions. However, the combination of SA and GA outperformed other algorithms with a significantly improved fitness score of 34, highlighting the synergy between these two approaches. PSO exhibited superior performance, yielding a fitness score of 29. This algorithm's ability to optimize scheduling in cellular manufacturing suggests its suitability for complex and dynamic manufacturing environments. The AMGA produced a respectable fitness score of 46, showcasing its capability to adapt and refine solutions iteratively. The algorithm's ability to adjust the mutation rate based on the problem complexity contributes to its effectiveness in scheduling optimization. SA combined with TS achieved a fitness score of 35, indicating a promising approach to address scheduling challenges. The combination of these two algorithms leverages the exploration and exploitation capabilities, leading to improved results. Although the Duelist Algorithm obtained a relatively high fitness score of 112, it demonstrated a less optimal performance compared to the other algorithms considered.

X. CONCLUSION

In conclusion, this paper presents a comprehensive analysis of scheduling optimization in cellular manufacturing using various meta-heuristic algorithms. Even though the DA has a high fitness value it converges quickly. Since cellular manufacturing units rely on the communication between individual units in the cells, it performs well when using swarm intelligence-based algorithm (PSO). A hybrid approach between SA and TS performs better than the GA. The same can be said about the hybrid approach between SA and GA. SA seems to be harmonious with most of the meta-heuristic algorithms. By adapting a non-constant mutation rate the Genetic algorithm is improvised further. By observing Fig. 1 it is obvious that this approach has explored the farthest ends of fitness value while looking for optimum value of fitness.

Further investigation and fine-tuning may be necessary to enhance its effectiveness in cellular manufacturing scheduling optimization. Approaches such as branch and limit and dynamic programming with increasing the number of tasks has limited computation time and limited storage on the computer; therefore, the use of innovative and meta-heuristic algorithms can be effective. Especially The hybrid approaches have outperformed the existing meta-heuristic algorithms. This proposed work only tries to solve scheduling issues for a small cellular manufacturing unit. This can be implemented for NP-hard problem datasets also. This is a single

objective optimization methodology but this can be optimized with other CMS problems using a Multi Objective optimization technique also.

REFERENCES

- [1] Ariafar, S., & Ismail, N. (2009). An improved algorithm for layout design in cellular manufacturing systems. *Journal of Manufacturing Systems*, 28(4), 132–139. <https://doi.org/10.1016/j.jmsy.2010.06.003>
- [2] Mahdavi, I., Paydar, M. M., Solimanpur, M., & Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications*, 36(3), 6598–6604. <https://doi.org/10.1016/j.eswa.2008.07.054>
- [3] Nouri, H., & Hong, T. S. (2013). Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations. *Journal of Manufacturing Systems*, 32(1), 20–31. <https://doi.org/10.1016/j.jmsy.2012.07.014>
- [4] Tavakkoli-Moghaddam, R., Gholipour-Kanani, Y., & Cheraghalizadeh, R. (2008). A genetic algorithm and memetic algorithm to sequencing and scheduling of cellular manufacturing systems. *International Journal of Management Science and Engineering Management*, 3(2), 119–130. <https://doi.org/10.1080/17509653.2008.10671040>
- [5] Solimanpur, M., Vrat, P., & Shankar, R. (2004). Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing. *European Journal of Operational Research*, 157(3), 592–606. [https://doi.org/10.1016/s0377-2217\(03\)00248-0](https://doi.org/10.1016/s0377-2217(03)00248-0)
- [6] O. Arghish et al., "An Integrated Cellular Manufacturing System with Type-2 Fuzzy Variables: Three Tuned Meta-heuristic Algorithms," *International Journal of Advanced Manufacturing Technology*, vol. 94, no. 1-4, pp. 2293-2308, Jan. 2018. <https://doi.org/10.1007/s00170-017-0762-3>.
- [7] B. Vizvári, H. Güden, and M. Ghadiri Nejad, "Local search based meta-heuristic algorithms for optimizing the cyclic flexible manufacturing cell problem," *Advances in Operations and Production Management*, vol. 13, no. 1, pp. 15-32, 2019. <https://doi.org/10.22121/aotp.2019.148030.1015>
- [8] Kesavan, R., Kamalakannan, R., Sudhakarapandian, P., Sivakumar, "Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review," in *Materials Today: Proceedings*, vol. 21, no. 1, pp. 66-72, 2020. <https://doi.org/10.1016/j.matpr.2019.05.363>
- [9] M. Fathi and M. Ghobakhloo, "Enabling Mass Customization and Manufacturing Sustainability in Industry 4.0 Context: A Novel Heuristic Algorithm for in-Plant Material Supply Optimization," *Sustainability*, vol. 12, no. 16, p. 6669, Aug. 2020, <https://doi.org/10.3390/su12166669>
- [10] A. Rezaeiapanah and M. Mojarad, "Modeling the Scheduling Problem in Cellular Manufacturing Systems Using Genetic Algorithm as an Efficient Meta-Heuristic Approach", *JAIT*, vol. 1, no. 4, pp. 228–234, Oct. 2021.
- [11] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [12] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, pp. 33–57, 2007. [Online]. Available: <https://doi.org/10.1007/s11721-007-0002-0>
- [13] S. Marsili Libelli and P. Alba, "Adaptive mutation in genetic algorithms," *Soft Computing*, vol. 4, pp. 76–80, 2000. [Online]. Available: <https://doi.org/10.1007/s005000000042>
- [14] T.R. Biyanto et al., "Duelist Algorithm: An Algorithm Inspired by How Duelist Improve Their Capabilities in a Duel," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and B. Niu, Eds. Cham: Springer, 2016, pp. 4–. [Online]. Available: https://doi.org/10.1007/978-3-319-41000-5_4